CHAPTER

12

SUPPORTING WINDOWS 9X

In this chapter, you will learn:

- About Windows installations and customizing the Windows environment
- How to install and resolve problems with applications software
- How to manage Windows resources, including memory and hard drives
- About ways to optimize Windows performance
- About the Windows registry and how to repair a corrupted registry
- How to use some diagnostic software
- About Plug and Play and how to troubleshoot Plug and Play problems

Supporting Windows requires a general knowledge of how hardware works and a detailed knowledge of how Windows and other types of software work. Having a conceptual understanding of how Windows works to manage memory, other hardware devices, and applications software helps in troubleshooting and problem solving.

Before discussing the details of supporting Windows 9x, this chapter explains the fundamental ways in which Windows 9x differs from DOS and Windows 3.x, the variety of software and hardware compatibility problems, and the nature of Windows NT and its successor, Windows 2000. The chapter then turns to ways to support Windows 9x, including installation, memory management, hard disk management, and general troubleshooting.

How Windows 9x Differs from Windows 3.x and DOS

Windows 9x is an OS that bridges two worlds. In Figure 12-1, you see that Windows 3.x and DOS constitute a 16-bit world with memory management centered around conventional, upper, and extended memory limitations. Windows 9x, as you have seen throughout several chapters, still has a DOS-based core, uses many 16-bit programs, and must manage base, upper, and extended memory in fundamentally the same way as does DOS. However, Windows 95 introduces 32-bit programming, dynamically loaded device drivers, memory paging, networking, and many other features available in Windows NT and Windows 2000. Windows 9x claims to be completely backward compatible with older software and with hardware designed to work in a DOS and Windows 3.x environment. Windows 9x uses cooperative multitasking when supporting 16-bit applications and preemptive multitasking when supporting 32-bit applications. Windows NT is the break with the past. It does not claim total backward compatibility, because it is a freshly designed OS with new ways of managing software and hardware resources.

A+ OS 1.1

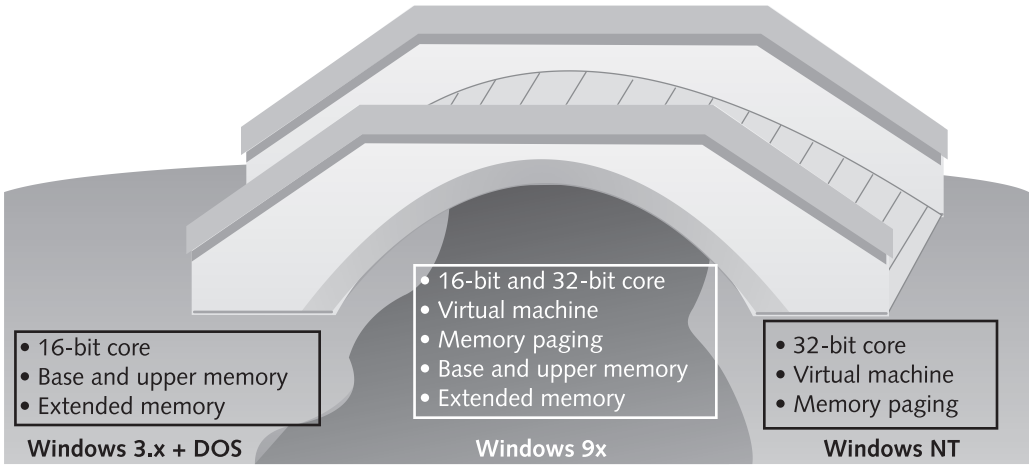


Figure 12-1 Windows 9x is the bridge from DOS to Windows NT

The Windows 9x Core

A+ OS 1.1

Windows 9x has at its core the same three components as Windows 3.x: the user, the kernel, and the GDI. The purposes of each component are listed in Table 12-1.

Figure 12-2 shows the three core portions of the Windows 9x OS. In Figure 12-2, you can see that the basic Windows 9x core component, the kernel, uses mostly 32-bit code. The 16-bit code is only retained as entry points into the kernel from 16-bit application programs. The user portion uses mostly 16-bit code, primarily because it uses less memory than the 32-bit equivalent and does not have a need for significant speed. The GDI core uses a mix of 16-bit and 32-bit code in order to maintain compatibility with 16-bit application programs.

A+ OS 1.1

Table 12-1 Core components of Windows 9x

Component Name	Main Files Holding the Component	Functions
Kernel	Kernel32.dll, Krnl386.exe	Handles the basic OS functions such as managing memory, file I/O, and loading and executing programs
User	User32.dll, User.exe	Controls the mouse, keyboard, ports, and desktop, including the position of windows, icons, and dialog boxes
GDI	GDI32.dll, GDI.exe	Draws screens, graphics, and lines, and prints them

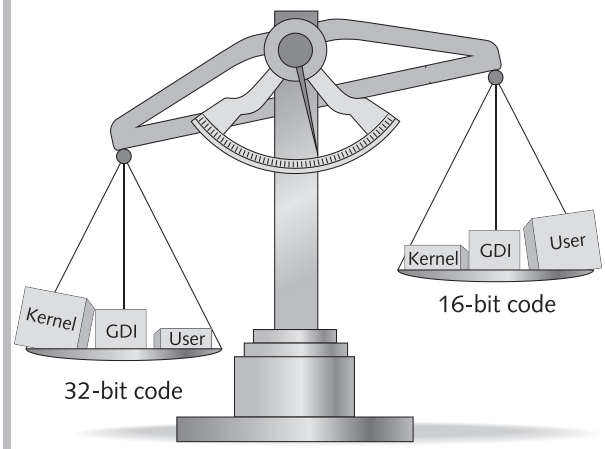


Figure 12-2 Windows 9x uses some 32-bit and some 16-bit code in its three core components

The Windows 9x Architecture

The core of Windows 9x consists of the three components described above. This core relates to users, software, and hardware by way of several satellite components, as seen in Figure 12-3. Just as DOS and Windows 3.x each provided a shell for the user to interface with the OS, Windows 9x provides a group of user interface tools and a shell for applications. Configuration data that was once stored in Windows 3.x .ini files is now stored in the Windows 9x registry, a database that also contains the initialization information for applications. There is more about .ini files and the registry later in the chapter.

A+ OS 1.1

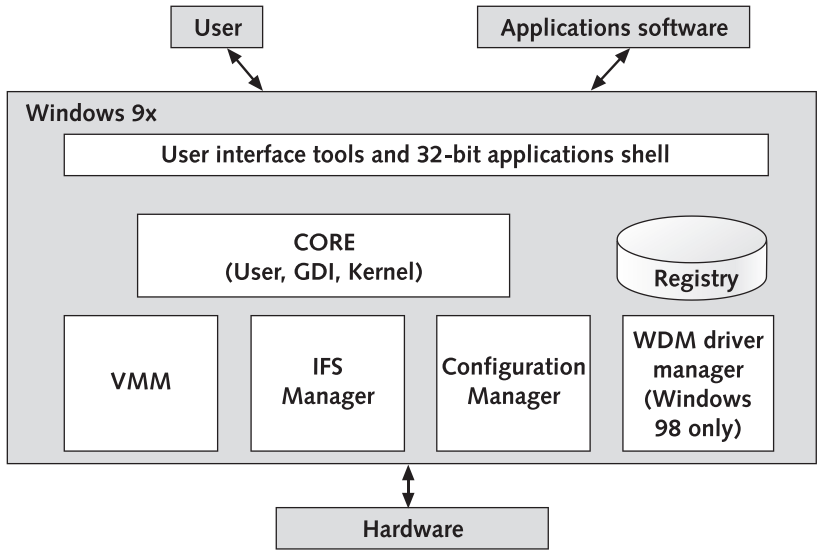


Figure 12-3 The Windows 9x architecture as it relates to the user, applications software, and hardware

One component manages memory used by other components and applications in a virtual machine environment that is similar to, but more sophisticated than that of Windows 3.x. This component is called the Virtual Memory Manager (VMM).

Another component, called the Installable File System (IFS) Manager, is responsible for all disk access. The Configuration Manager is responsible for the Plug and Play features of Windows 9x and other hardware configuration tasks. In Figure 12-3, the only component that is found in Windows 98, but not Windows 95 is the component responsible for managing device drivers that work under a driver model new to Windows 98, called Win32 Driver Model (WDM) driver manager.

Figure 12-3 serves as a simple, but complete reference point for all the components of Windows 9x, illustrating how they relate to the user, hardware, and software, and to each other. As you can see, Windows 9x architecture uses a modular approach. Keep in mind that Windows 9x is the compromise OS, attempting to bridge two worlds which can, in a simplistic way, be defined as a 16-bit world and a 32-bit world.

16-Bit and 32-Bit Programming

DOS is a 16-bit OS. All portions of the OS are written using 16-bit code, and DOS only supports 16-bit drivers and 16-bit application programs. Windows 3.x also only supports 16-bit applications, but does contain a small amount of 32-bit code. Recall from earlier chapters that Windows for Workgroups 3.11 introduced fast 32-bit programming for disk access, which we now call VFAT, but generally speaking Windows 3.x is a 16-bit program and only supports 16-bit application programs.

Windows 9x contains some 16-bit code and some 32-bit code. Programs written in 32-bit code require more memory and are generally faster than programs written in 16-bit code. Windows 9x supports VFAT, which is written with 32-bit code. Although Windows 9x supports 16-bit device drivers, it's preferable to use the 32-bit drivers supplied with Windows 9x for three main reasons. They are generally much faster, and 32-bit drivers can be stored in extended memory, releasing more of the first MB of memory to application programs. Also, 32-bit drivers can be dynamically loaded, meaning that they are loaded into memory when they are needed and then removed when not needed, thus conserving memory. In contrast, 16-bit drivers must be stored in conventional or upper memory. When Windows 9x is installed over DOS, it searches for these 16-bit drivers and replaces them with 32-bit drivers if it can.

Virtual Machines

A+ OS 1.1, 2.4, 3.1

Another important difference between Windows 9x and DOS with Windows 3.x is that Windows 9x enhances the application of virtual machines. Think of **virtual machines** (VM) as several logical machines within one physical machine, as represented in Figure 12-4, similar in concept to several logical drives within one physical hard drive.

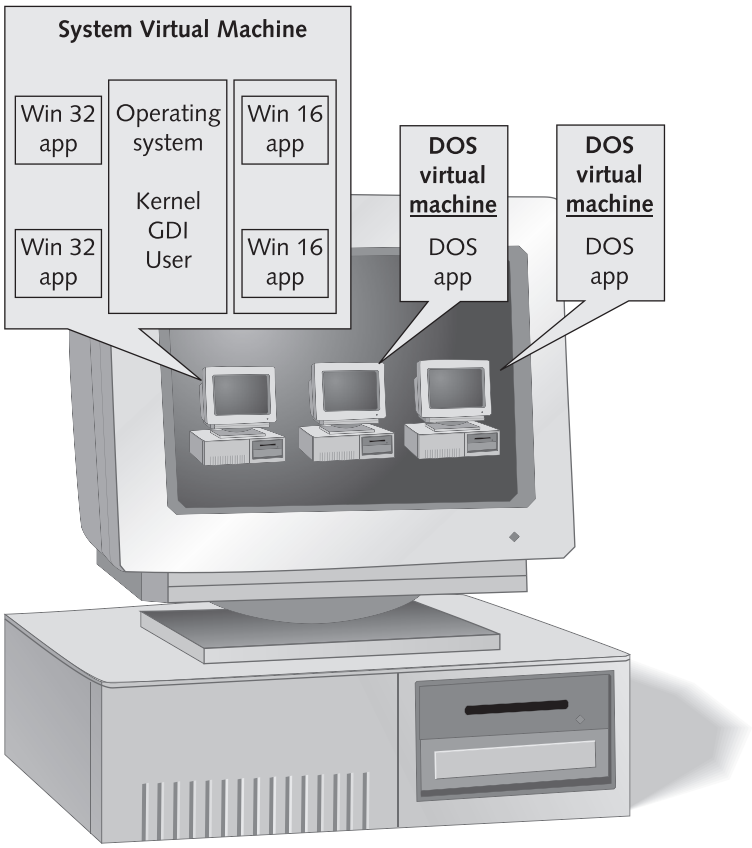


Figure 12-4 Windows 9x uses the virtual machine concept

A+ OS 1.1, 2.4, 3.1

In Figure 12-4, the system Virtual Machine can support 32-bit and 16-bit Windows application programs, but DOS programs are put aside into their own virtual machines. Remember that DOS programs don't account for the sharing of resources. A DOS program expects to directly control the hardware of the entire PC, memory included. If a DOS program begins to use memory addresses not assigned to it, errors occur in a multitasking environment. Windows 9x solves this problem by providing the DOS program with its own logical machine. In effect, the application program says, "I want all of memory and all of this and all of that." Windows 9x says, "OK, here they are," and gives the program its own PC, including all the virtual memory addresses it wants from 0 to 4 GB as well as its own virtual hardware! As far as the DOS program is concerned, it can go anywhere and do anything within its own PC. That's a virtual machine! The DOS application program does not try to communicate with any other application program or to access the data of another program, because it thinks there are no other programs—it controls its entire world, and it's the only program in it.

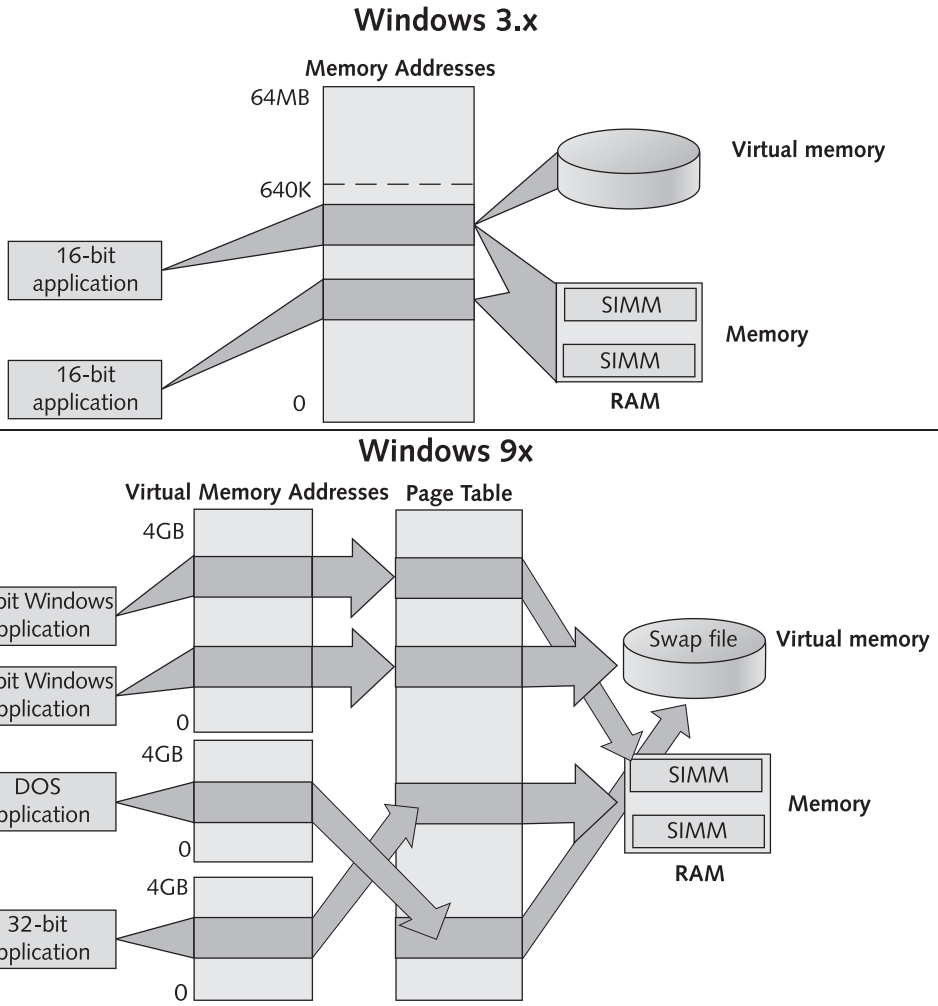
One important result of running DOS programs in their individual virtual machines is that when a program makes an error, the virtual machine it is using hangs, but the rest of the actual PC is isolated from the problem.

Windows 16-bit application programs offer a slightly different challenge to Windows 9x. These programs make some of the same mistakes that DOS programs do and can cause the system to hang. However, they also sometimes expect to access other programs and their data. The 16-bit Windows programs don't expect to control the hardware directly, and are content to route their requests to Windows. Windows 9x places these programs within the system VM because they communicate with hardware through the OS, but Windows 9x puts these programs together in their own memory space so they can share memory addresses.

The result of this arrangement is that, when a 16-bit Windows program causes an error called a Windows Protection Error or a General Protection Fault, it can disturb other 16-bit programs, causing them to fail, but it does not disturb DOS programs in their own VM or 32-bit programs that don't share their virtual memory addresses.

Memory Paging

How does Windows 9x provide virtual memory addresses to DOS and 16-bit Windows application programs? By **memory paging** which is managed by the **Virtual Memory Manager**. Look at Figure 12-5. In the top diagram, you see Windows 3.x with the memory model you have observed in earlier chapters. Application programs in Windows 3.x share the memory addresses that have been assigned to either the physical or virtual memory of a system. For example, in Figure 12-5, 64 MB of memory addresses are available. Although 16-bit programs run in conventional memory (first 640K), they might store their data in extended memory. Of the 64 MB, perhaps half of these addresses are assigned to physical RAM stored on SIMMs or DIMMs, and the other half of the addresses are virtual memory contained in the swap file on the hard drive. In this case, there is only one set of memory addresses, and all application programs must share these addresses.



A+ OS 3.2, 1.1

Figure 12-5 How Windows 9x manages memory differently than does Windows 3.x

As you can see in the lower part of Figure 12-5, Windows 9x not only has virtual memory stored in a swap file, but also provides virtual memory addresses to application programs. In Figure 12-5, you see three sets of virtual memory addresses. Each set can contain 0 to 4 GB of addresses, depending on the amount of virtual memory available. The top set is being used by two 16-bit programs. The second set of virtual addresses is being used by a single DOS program, and a third set of addresses is being used by a 32-bit program. Each VM for DOS has a set of virtual memory addresses. The 16-bit Windows programs share a single set of virtual memory addresses, and each 32-bit program has its own individual set of addresses.

A+ OS 3.2, 1.1

In Figure 12-5, all these virtual addresses map onto the page table, which in turn maps onto either physical memory (RAM) or virtual memory on the hard drive (swap file). Obviously, not all virtual memory addresses in Windows 9x have physical or virtual memory assigned to them. These virtual addresses remain unassigned until an application program uses them.

In Windows 9x, the Virtual Memory Manager controls the page table, moving 4K pages in and out of physical RAM. If a program requests memory that the memory manager knows is stored in the swap file, the manager generates a **page fault**, which causes the manager to go to the drive to return the data from the swap file to RAM. This action is called a **page-in**. If RAM is full, the manager takes a page and moves it to the swap file, which is called a **page-out**.

If RAM is full much of the time, the Virtual Memory Manager might spend excessive time moving pages in and out of RAM, which can cause excessive hard drive use and a decrease in overall system performance and can even cause a system lockup or applications to fail. This situation is sometimes called **disk thrashing** and can cause premature hard drive failure. Symptoms of excessive memory paging are:

- Very high CPU use
- Very slow system response
- Constant hard drive use

The solution is to leave fewer application programs open at the same time or to install more RAM.

Having covered the differences between Windows 95 and Windows 3.x with DOS, the chapter now briefly looks at how Windows 98 differs from Windows 95. The differences are much less significant, as Windows 98 is really just an upgrade from Windows 95.

How Windows 98 Differs from Windows 95

Windows 98 is basically the same operating system as Windows 95, having the same basic core components and providing the same fundamental services to software, hardware, and the user. However, it does offer some added features and improved performance over Windows 95. Some of these features, including FAT32 and support for USB and DVD, became available with Windows 95, Release 2. Windows 98 also includes some new system tools to monitor and improve system performance, new hardware support, and additional Web tools. Table 12-2 summarizes the changes from Windows 95 to Windows 98.

FAT32 is not compatible with Windows NT or disk compression, including DoubleSpace under Windows 98. To use disk compression or to have a file system compatible with Windows NT, use FAT16. Windows 2000 supports FAT32.

Windows 98 and Window 95 load, run, and install very much the same way. As we continue through this chapter, any significant differences between the two versions will be noted.

Table 12-2 Features new to Windows 98

Feature	Description
Troubleshooting utilities	Windows 95 had a few troubleshooting utilities, but the 15 utilities that come with Windows 98 are more interactive.
Update Wizard	The Update Wizard connects to the Microsoft Web site and automatically downloads any new drivers or fixes.
Maintenance Wizard	The Maintenance Wizard can be used to regularly schedule several maintenance tasks, which include running Disk Defragmenter and ScanDisk, discussed in earlier chapters.
DriveSpace 3	An improved version of DriveSpace for Windows 95, it includes a third level of data compression, called UltraPack which takes up less space per file than does regular compression, called HiPack.
Power management support	Windows 98 supports some power management features, if both hardware and software are present to use them.
Registry Checker	Backs up and restores the registry
Web tools and features	Several Windows 98 features take on an Internet look and feel. Windows 98 also supports viewing TV and interactive programs. You'll need a special TV interface card to do it.
FAT32	Recall that FAT32 is a file system that allows for a smaller cluster size on large drives than the earlier FAT16.
New hardware support	With 1,200 device drivers, Windows 98 supports many more hardware devices than did Windows 95. Also, Windows 98 supports DMA channels for IDE CD-ROM drives, USB, DVD, and multiple video cards supporting multiple monitors.
Win32 Driver Model (WDM)	A new device driver model, also used by Windows NT, makes it possible for the same driver to be used by both operating systems.

Windows 98 Upgrades

A+ OS 2.2

Microsoft has produced two upgrades for Windows 98: Windows 98 Second Edition (Windows 98 SE) and Windows Millennium Edition (Windows Me). Each upgrade has significant enhancements over its predecessor.

Windows 98 SE includes several patches, or fixes, for the first edition of Windows 98, updates of existing components, and some new components. Most new features involve networking and Internet access. Improved support for ATM networks includes the addition of Point-to-Point Protocol (PPP) over ATM, which allows a dial-up connection over an ATM network. Security for a dial-up connection over regular phone lines was also upgraded.

A new feature is Internet Connection Sharing (ICS), which makes it possible for a Windows 98 PC to access the Internet through another computer on a local network so that only one computer requires a direct connection to an ISP. This feature means that

several PCs on a small home network can share the same access to an ISP without incurring additional charges and without installing third-party software. Support for modems that use a USB port and support for a wake-on-LAN connection are also included. Wake-on-LAN means that a PC can go into a low-power state and then return to standard power when the network card detects activity from another computer on the network.

Windows Me takes us one step closer to the merging of Windows 9x and Windows 2000 as it contains features from each OS, although, at its core, it is still a Windows 9x upgrade. It's designed for the home user and not for business. It focuses on enhancements to multimedia features such as support for video cameras, digital cameras, scanners, and a jukebox recorder. It includes a compression utility for video files and a video editor. True to its goal as a home PC operating system, the OS is very user friendly, including more informative error messages and troubleshooting utilities. Other features unique to Windows Me are discussed at the end of this chapter.

A+ OS 1.1

To know which version of Windows is installed, right-click on the My Computer icon and select **Properties** from the shortcut menu. The System Properties windows opens. Click the **General** tab.

Loading and Running Windows 9x

We now turn our attention to what happens when Windows 9x starts up, how a user can alter the startup process, and how the Windows 9x desktop can be customized. With that foundation built, we will then turn our attention to installing Windows 9x, and continue on to a more in-depth look at support issues.

Files Used to Customize the Startup Process

A+ OS 1.1

Before you learn about the startup process, you will learn about the files that Windows 9x uses to control the process. Recall from earlier chapters that DOS requires IO.SYS, MSDOS.SYS, and COMMAND.COM in the root directory of the boot device to load. In addition, AUTOEXEC.BAT and CONFIG.SYS are text files that can contain settings for environmental variables and commands to load drivers and TSRs. If AUTOEXEC.BAT or CONFIG.SYS files are present in the root directory, the command lines in them are executed during the boot. They are used to customize the load. Just as DOS uses text files to contain information about the load, Windows 3.x also uses text files to hold custom settings that help control the loading process. These files are called initialization files, and some of the entries in these files are read and used by Windows 9x. However, most of Windows 9x settings are stored in a database called the Windows registry rather than in text files. You will learn about the Windows registry later in the chapter, but now we turn our attention to how initialization files are organized and how they can be used. The Windows System Configuration Editor (Sysedit) can be used to edit text files. To use Sysedit, type SYSEDIT in the Run dialog box. These files automatically display for editing: Autoexec.bat, Config.sys, Win.ini, and Protocol.ini.

A+ OS 1.1

Initialization Files

An **initialization file**, with an .ini file extension, is a file used by Windows or applications software to store configuration information needed when Windows or an application is first loaded. An application can have its own .ini files and registry and can also store its information in the Windows .ini files and the Windows registry. The contents of files with an .ini extension are organized into sections, which are each given a name. Within a section, values are assigned to variables using this format:

```
[SECTION NAME]
KEYNAME=value
```

Any value to the right of the keyname becomes available to Windows or an applications software program reading the file; in other words, the keyname acts much like the SET command in AUTOEXEC.BAT, which assigns a value to a system variable.

System.ini and Win.ini are used by both Windows 3.x and Windows 9x. A sample Windows 9x System.ini file is shown in Figure 12-6. The two sections that have the most impact on the boot process are [boot] and [386Enh]. Windows 3.x keeps many more entries in these sections than does Windows 9x, which really only uses these files for backward compatibility with older applications.

.ini files are only read when Windows or an application using .ini files starts up. If you change the .ini file for an application, you must restart the software for the change to take effect. If you want the application to ignore a line in the .ini file, you can turn the line into a **comment line**, which is ignored by the software and is used to document the file. To make a line a comment line, put a semicolon at the beginning of the line.

The maximum file size of .ini files is 64K, although files greater than 32K can cause some applications software problems. Most applications have a setup program in their program group that is used to make changes to their .ini files.

Sometimes it is necessary to manually edit an .ini file that belongs to an application, but you should not edit System.ini or other Windows 9x initialization files because Windows might not run correctly, and because Windows might overwrite these files when changes are made to Windows through the Control Panel.

We now turn our attention to studying the Windows 9x startup process.

The Windows 9x Startup Process

A+ OS 1.1, 2.3

Windows 9x first loads in real mode and then switches to protected mode. With DOS, the two core real-mode system files responsible for starting up the OS, IO.SYS and MSDOS.SYS remain in memory, running even after the OS is running. With Windows 9x, Io.sys is responsible for only the initial startup process performed in real mode, then control is turned over to Vmm32.vxd, which works in protected mode, and Io.sys is terminated. Windows 9x includes a file named **Msdos.sys**, but it is only a text file that contains some parameters and switches that can be set to affect the way the OS boots.

A+ OS 1.1, 2.3

Starting up Windows 9x is a five-phase process, as shown in Figure 12-7. We will next look at each phase in turn.

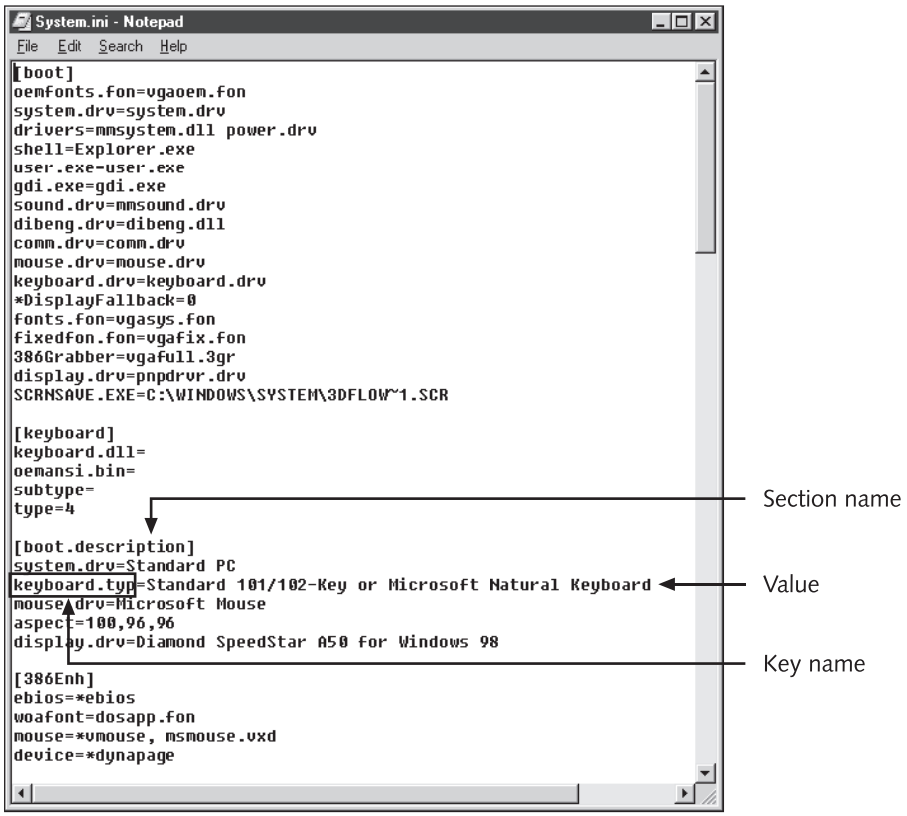


Figure 12-6 A sample Windows 98 System.ini file

A+ OS 1.1, 2.3

Phase 1: BIOS Bootstrap Startup BIOS begins the process. If the BIOS is **Plug and Play (PnP) BIOS**, then it looks to permanent RAM for information about hardware and configures PnP devices that have their configuration information recorded there. It performs POST and saves information that Windows Configuration Manager later uses to complete the hardware configuration.

Phase 2: DOS Drivers, TSRs and Environmental Settings In Phase 2, BIOS turns control over to Io.sys, which creates a real-mode operating system environment. It automatically loads several drivers, sets several environmental variables, and executes any commands listed in Config.sys and Autoexec.bat. Config.sys and Autoexec.bat are not used in the same way that they were with DOS, because many of their commands are automatically executed by Io.sys. However, for backward compatibility, any entries in these files are executed.

A+ OS 1.1, 2.3

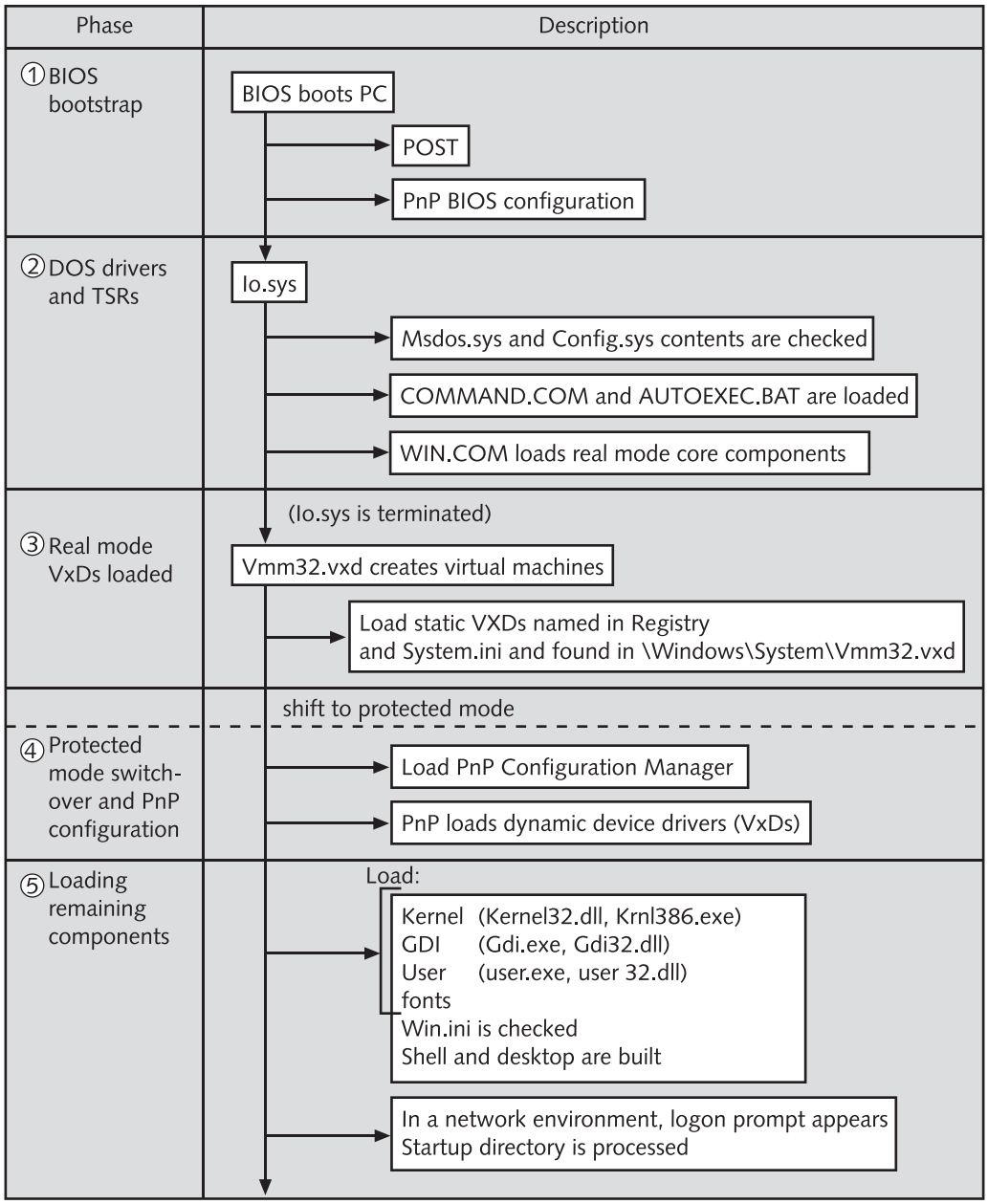


Figure 12-7 Windows 9x core components and the loading process

A+ OS 1.1, 2.3

Io.sys checks the text file Msdos.sys for boot parameters, and automatically loads the following drivers if they are present: Himem.sys, Ifshlp.sys, Setver.exe, and Drvspace.bin (or Dblspace.bin). Himem.sys provides access to extended memory. Ifshlp.sys is used by 16-bit programs to access the file system. Setver.exe is included for backward compatibility with DOS applications that use the DOS version number. Drvspace.bin or Dblspace.bin provide disk compression. One of these two files is loaded only if Io.sys finds Dlbspace.ini or Drvspace.ini in the root directory of the boot drive. Io.sys does not load Emm386.exe. Therefore, if you have a 16-bit program that requires emulated, expanded memory, then load Emm386.exe from the Config.sys file.

Io.sys also sets several environmental variables to default settings. Entries in Io.sys cannot be edited, but an entry in Config.sys overrides the default entry in Io.sys. Therefore, if you want to use settings different from the default, put the command in Config.sys. Here are the default Io.sys entries: files=60, lastdrive=z, buffers=30, stacks=9,256, shell=Command.com, and Fcbs=4.

Next, Io.sys loads Command.com, and follows instructions stored in Autoexec.bat. The default assignments made to environmental variables that were stored in Autoexec.bat in DOS are listed below:

- Tmp=c:\windows\temp
- Temp=c:\windows\temp
- Prompt=pg
- Path=c:\windows;c:\windows\command

You can change any of these by making an entry in Autoexec.bat. Next, Io.sys loads Win.com. Then Win.com loads other real-mode core components.

Windows 9x does not need CONFIG.SYS and AUTOEXEC.BAT, but executes them only to provide backward compatibility for older hardware and software. Settings that were once included in CONFIG.SYS and AUTOEXEC.BAT are now built into Io.sys and the registry. However, if you put command lines in either of these two files, they are executed and override the settings stored in Io.sys and the registry.

Phase 3: Static VxDs In Phase 3, Io.sys relinquishes control to the Virtual Memory Manager component housed in Vmm32.vxd along with some VxDs. Recall that a VxD is a virtual device driver that works with a virtual machine to provide access to hardware for software running in the VM. Under Windows 3.x, these VxDs were loaded from System.ini and had a 386 file extension. Under Windows 9x, if stored in individual files, they have a .vxd file extension. They are called **static VxDs** because once they are loaded into memory, they remain there. (Conversely, **dynamic VxDs** are loaded into and unloaded from memory as needed.)

Vmm32.vxd was built specifically for this computer when Windows 9x was installed and contains some VxDs critical for a successful boot. (The VxD drivers that are now included in Vmm32.vxd were listed in the [386enh] section of System.ini under Windows 3.x.)

A+ OS 1.1, 2.3

Vmm32.vxd terminates Io.sys and, while still in real mode, loads static VxD device drivers as identified in four different locations. They can be embedded in Vmm32.vxd, named in the registry or System.ini, or stored in the VxD file in the \Windows\System\Vmm32 directory.

If you suspect a problem with a VxD that is part of the Vmm32.vxd file, then store a new version of the file in the \Windows\System\Vmm32 directory. If Windows finds a VxD driver there, it uses that driver instead of the one embedded in Vmm32.vxd. Also, VxD drivers are listed in the registry and also listed in System.ini. Normally, the entries are the same, and entries in System.ini are only listed there for backward compatibility. However, if an entry in System.ini differs from an entry in the registry, the value in System.ini is used.

Phase 4: Protected Mode Switchover and PnP Configuration At the beginning of Phase 4, Vmm32.vxd switches to protected mode and loads Configuration Manager. Configuration Manager is responsible for configuring legacy and PnP devices. It will use any information that PnP BIOS might have left for it, and loads the 32-bit dynamic device drivers (VxDs) for the PnP devices.

Phase 5: Loading Remaining Components In Phase 5, with Vmm32.vxd still in control, the three core components are loaded, fonts and other associated resources are loaded. Win.ini is checked, and commands stored there are executed to allow for backward compatibility. The shell and user desktop are loaded. If the computer is working in a networked environment, a logon dialog box is displayed, and the user can log on to Windows 9x and the network. Finally, any processes stored in the Startup directory are performed.

Microsoft Windows 9x Startup Menu

A+ OS 3.1

Normally, when you load Windows, the message, *Starting Windows 98* displays and then the OS loads. However, if you have set up your PC for a dual boot (how to do that was covered in Chapter 7), the following menu is always displayed when you boot. The time that the menu stays on the screen is determined by a setting in Msdos.sys. You can force the menu to display rather than the *Starting Windows 98* message by holding down either the F8 key or the Ctrl key during the boot. The Microsoft Windows 9x Startup Menu options are:

1. Normal
2. Logged (\BOOTLOG.TXT)
3. Safe mode
4. Safe mode with network support
5. Step-by-step confirmation
6. Command prompt only
7. Safe mode command prompt only
8. Previous version of MS-DOS

What to expect when you select each option on the menu is described next.

Option 4 is displayed if the OS is configured for a network, and Option 8 is displayed if a previous version of DOS was retained during the Windows 9x installation.

Normal In Msdos.sys, if BootGUI=1, then this option starts Windows 9x. If BootGUI=0, then this option will boot to the DOS 7.0 or DOS 7.1 prompt. Either way, the commands in AUTOEXEC.BAT and CONFIG.SYS will be executed.

Logged (\BOOTLOG.TXT) This option is the same as Normal, except that Windows 9x tracks the load and startup activities and logs them to the bootlog.txt file. This file can be a helpful tool when troubleshooting.

A+ OS 2.3, 3.1

Safe mode Safe mode starts Windows 9x with a minimum default configuration to give you an opportunity to correct an error in the configuration. Windows 98 includes support for networks, but Windows 95 does not. For example, if you selected a video driver that is incompatible with your system, when Windows 9x starts, it detects the problem and enters Safe mode with a standard VGA driver selected. You can then go to Device Manager, select the correct driver, and restart Windows.

From the Startup menu, you can choose to enter safe mode yourself if you know of a problem you want to correct. For example, if you had previously selected a group of background and foreground colors that makes it impossible to read the screens, you can reboot and choose Safe mode. Safe mode gives you the standard color scheme along with the VGA mode. Go to Display Properties, make the necessary corrections, and reboot.

In safe mode, the commands in AUTOEXEC.BAT and CONFIG.SYS are not executed.

To load Windows 9x in safe mode, press F5 when the message "Starting Windows 95/98" appears.

Safe mode with Network Support This option allows access to the network when booting into safe mode. It is useful if Windows 95 is stored on a network server, and you need to download changes to your PC in safe mode. This option is not available in Windows 98.

Step-by-Step Confirmation The option asks for confirmation before executing each command in Io.sys, CONFIG.SYS, and AUTOEXEC.BAT. You can accomplish the same thing by pressing Shift+F8 when the message "Starting Windows 95/98" appears.

A+ OS 1.1, 2.3

Command Prompt Only This option executes the contents of AUTOEXEC.BAT and CONFIG.SYS, but doesn't start Windows 9x. You will be given a DOS prompt. Type **WIN** to load Windows 9x. Type WIN to load Windows 9x, which executes the file WIN.COM. When troubleshooting the OS, try these switches with the WIN command: WIN/D:F turns off 32-bit disk access and WIN/D:M starts Windows in Safe Mode.

Safe mode Command Prompt Only This option does not execute the commands in AUTOEXEC.BAT or CONFIG.SYS. You will be given a DOS prompt.

Previous Version of MS-DOS This option loads a previous version of DOS if one is present. You can get the same results by pressing F4 when the message "Starting Windows 95/98" appears. This option is not available in Windows 98 SE.

Keystroke Shortcuts When Navigating Windows

Table 12-3 lists a few handy keystrokes to use when working with Windows. The function keys used during startup are included to make the table complete. You can also use the mouse to accomplish some of these same tasks, but keystrokes are faster for experienced typists. Also, sometimes in troubleshooting situations, the mouse is not usable. At those times, knowing the keystrokes is very valuable. When using dual monitors, use the mouse to control which screen is active by moving from screen to screen.

Table 12-3 Keystrokes to help make navigating Windows easier

General Action	Keystrokes	Description
While loading Windows	F4	Load previous version of DOS
	F5	Start in Safe Mode
	F8	Display Startup Menu
	Shift + F8	Step-by-Step Confirmation
Working with text anywhere in Windows	Ctrl + C Ctrl + Ins	Shortcut for Copy
	Ctrl + A	Shortcut for selecting all text
	Ctrl + X	Shortcut for Cut
	Ctrl + V Shift + Ins	Shortcut for Paste
	Shift + arrow keys	Hold down the Shift key and use the arrow keys to select text, character by character
Managing programs	Alt + Tab	While holding down the Alt key, press Tab to move from one loaded application to another.
	Ctrl + Escape	Display the Start menu
	Alt + F4	Close a program window, or, if no window is open, shut down Windows
	Double-click	Double-click an icon or program name to execute the program.
	Ctrl + Alt + Del	Display the Task List, which you can use to switch to another application, end a task, or shut down Windows

To control display settings including how the screen displays and what screen saver pattern is used, right click anywhere on the desktop and select Properties from the shortcut menu. To create a screen saver password, click the Screen Saver tab. Select a screen saver and check Password protected. To set the password, click the Change button and then enter and confirm a password. Click OK to save your password and then click OK to exit the Display Properties window.

Table 12-3 Keystrokes to help make navigating Windows easier (continued)

General Action	Keystrokes	Description
Selecting items	Shift + click	To select multiple entries in a list (such as filenames in Explorer), click the first item and then hold down the Shift key and click the last item you want to select in the list. All items between the first and last are selected.
	Ctrl + click	To select several items in a list that are not listed sequentially, click the first item to select it. Hold down the Ctrl key as you click other items anywhere in the list. All items you have clicked on are selected.
Using menus	Alt	Press the Alt key to activate the menu bar.
	Alt, letter	After the menu bar is activated, press a letter to select a menu option. The letter must be underlined in the menu.
	Alt, arrow keys	After the menu bar is activated, use the arrow keys to move over the menu tree.
	Alt, arrow keys, enter	After the menu bar is activated and the correct option is highlighted, press Enter to select the option.
	Esc	Press Escape to exit a menu without making a selection.
Managing the desktop	Print Screen	Copy the desktop into the Clipboard
	Ctrl + Esc	Display the Start menu and moves the focus to the menu. (Use the arrow keys to move over the menu.)
	Alt + M	After the focus is on the Start menu, minimizes all windows and moves the focus to the desktop
Using the Windows key	WIN + E	Start Windows Explorer
	WIN + M	Minimize all windows
	WIN + Tab	Move through items on task bar
	WIN + R	Display the Run dialog box
	WIN + Break	Display the System Properties window

Managing the Windows 9x Desktop

From the Windows 9x desktop you can make applications automatically load at startup, create shortcuts to files and applications, and make the environment more user-friendly. You can hide and unhide the taskbar at the bottom of the desktop. To do that, click Start, Settings, Taskbar and Start Menu. Then click the Taskbar Options tab and select Auto hide.

A shortcut on the desktop is an icon that points to a program that can be executed. The user double-clicks the icon to load the software. A shortcut can be created in several ways. One way is to use the Properties option on the taskbar. Right-click the taskbar and select

Properties from the menu that appears (see Figure 12-8). From this window, you can create a shortcut for a program or data file, name it, and select where to place it (either on the desktop or in the Start menu). If you want a program to load whenever Windows 9x starts, create a shortcut and put the shortcut in the StartUp folder of the Start menu. All items in the StartUp folder are automatically executed when Windows 9x starts.

To edit a shortcut, right-click the shortcut and select **Properties** from the menu. To delete a shortcut, select **Delete** from this same menu.

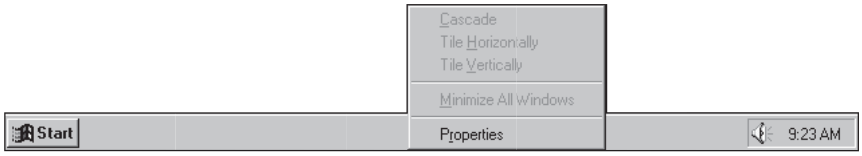


Figure 12-8 To customize the desktop, use the Properties sheet of the Windows 9x taskbar

Troubleshooting Icons on the Desktop An icon on the desktop can be a shortcut to an application or it can represent a file that belongs to an application. The telltale sign of the difference is the small shortcut symbol on the icon, as seen in Figure 12-9. The icon on the right represents the document file MyLetter1.doc stored in the \Windows\Desktop folder, and the icon on the left is a shortcut to the file MyLetter2.doc, which can be stored anywhere on the drive. Also shown in Figure 12-9 are the contents of the \Windows\Desktop folder as seen by Explorer. You can add an icon to the desktop by putting a file in this folder. One way to delete an icon on the desktop is to delete the corresponding file in this folder; however, as you will see, this method can cause problems for the user.

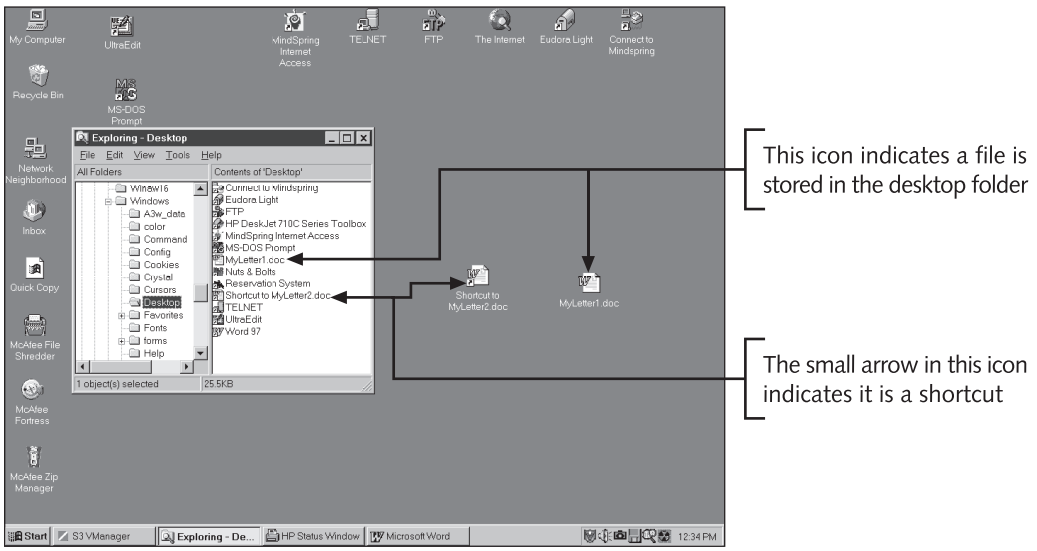


Figure 12-9 One icon is a shortcut, and the other icon represents a file stored in the Desktop folder

In the screen shown in Figure 12-9, it's possible to change the name of either icon so that they read the same on the desktop (to do this, click the icon name, which produces an insertion point), but the meaning of the two icons is still different. If you delete the icon on the right (the document icon), you have deleted the document file, MyLetter1.doc. If you delete the icon on the left (the shortcut), you have only deleted the shortcut, but the document file that it refers to, MyLetter2.doc, still exists somewhere on the drive.

An error can occur if the document file, MyLetter2.doc, is deleted, but the shortcut to the deleted document remains on the desktop. Figure 12-10 shows a sample error message that occurs when this shortcut is used.

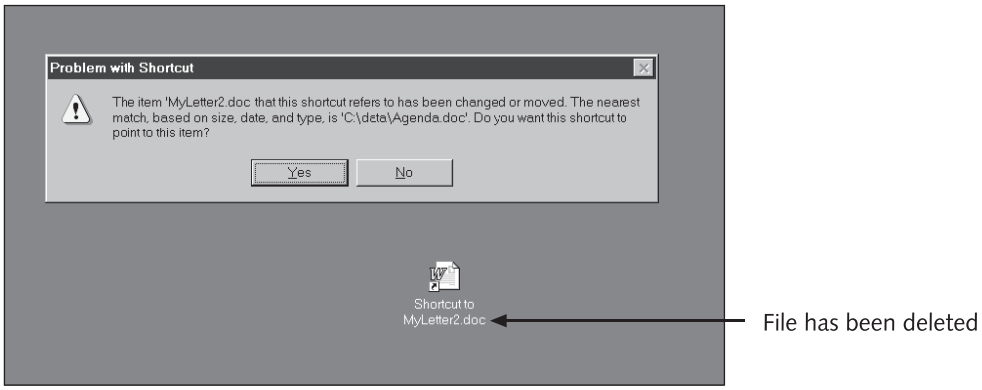


Figure 12-10 The file that the shortcut points to has been deleted, which causes an error when the shortcut is used

Explorer or Program Manager

Windows 3.x uses Program Manager as its primary tool to manage applications and OS components. Program Manager in Windows 3.x is executed by running Progman.exe. Windows 9x uses Explorer as its primary tool for managing applications and OS components. Explorer is executed by running Explorer.exe. Some people who are accustomed to Windows 3.x prefer to use Program Manager instead of Explorer. For these users, you can place a shortcut icon on the desktop to Progman.exe.

Installing and Configuring Windows 9x

Before installing Windows 9x, verify that the minimum requirements for the hardware are met. In order for Windows 9x to perform satisfactorily, the PC should meet the recommended requirements. Minimum and recommended requirements for Windows 95 and Windows 98 are listed below:

Minimum and recommended requirements for Windows 95:

- 486DX, 25 MHz or higher processor

- 4 MB of RAM (8 MB is recommended)
- 40 to 45 MB of hard disk storage, depending on the installation

Minimum and recommended requirements for Windows 98:

- 486DX, 66 MHz or higher processor
- 24 MB of RAM (32 MB is recommended)
- From 140 to 315 MB of hard disk storage, depending on the installation

Preparing for the Installation

A+ OS 2.1

Windows 9x is generally installed as an upgrade to DOS with Windows 3.x, as an upgrade from Windows 95 to Windows 98, or on a clean hard drive. If you have been having problems with the current operating system and applications, consider doing a clean install rather than an upgrade. A clean install requires that you reinstall all applications software after Windows is installed. To do that, first verify that you have all the applications software installation CDs or floppy disks and then back up all data on the drive. Also take the time to verify that the backup of data is good. If you like, you can also format the hard drive. Also, if you suspect a boot sector virus is present, use the FDISK/MBR command discussed in earlier chapters to rewrite the master boot sector program. Then do the clean install.

After Windows 9x is installed, reinstall all the applications software and then, if you formatted the hard drive, restore the data from backups. This method takes longer than an upgrade, but you get the added advantage of a "fresh start" and any problems with corrupted applications or system settings will not follow you into the new installation.

If you are doing an upgrade, before you begin the installation, prepare your hard drive by doing these things:

- Verify that you have enough space on the hard drive. Delete files in the Recycle Bin and temporary directories.
- Run SCANDISK or CHKDSK/F, discussed in Chapter 6, to recover lost clusters, delete the .CHK files created by these programs, and run DEFRAG to defragment your hard drive, which improves overall hard drive performance and possibly allows for a larger swap file.
- Run a current version of antivirus software to check for viruses.
- To ensure that you can backtrack to an earlier version of Windows, save AUTOEXEC.BAT and CONFIG.SYS, found in the root directory, and all files with .ini or .grp file extensions in the \Windows directory to a disk. Files with the .grp file extension are **group files**, which contain information about a program group displayed in Program Manager.

If you have problems with a Windows 9x installation, you can reinstall the original version of Windows 3.x and copy the .ini files and group files back to the \Windows directory to fully recover from the problem installation.

A+ OS 2.1

- Check CONFIG.SYS and AUTOEXEC.BAT for potential problems. Before you install Windows, optimize your memory with the methods discussed in Chapter 4, using Memmaker or manually editing these two files.
- If there are TSRs such as QEMM386 (a memory manager by Quarterdeck) loaded from CONFIG.SYS or AUTOEXEC.BAT, and problems arise because they are running during the installation, disable them by converting these lines to comments by typing **REM** at the beginning of the command lines. Later, after the installation, you can activate them again.
- If you are connected to a network, verify that the connection is working. If it is, then Windows setup should be able to reestablish the connection correctly at the end of the installation.
- If you are upgrading from Windows 95 to Windows 98, create a Windows 95 rescue disk for use in the event the installation fails.
- If you are installing Windows on a compressed drive, be aware that the registry can reside on any compressed drive, but the swap file can only reside on a compressed drive that is compressed using protected-mode software such as DriveSpace. DriveSpace marks the area for the swap file as uncompressible. If your drive is compressed with real-mode compression software, such as DoubleSpace, then you should put the swap file on another drive, possibly the host drive. If the host drive is not large enough to hold the file, check the software used to compress the drive, to see if you can increase the size of the host drive.

If you used DOS DoubleSpace (available with DOS 6 and 6.2) to compress the drive, type DBLSPACE at the DOS prompt. If you used DriveSpace (available with DOS 6.22), type DRVSPACE. The utilities will give you statistics about the compressed drive and also allow you to change the size of the host drive.

A+ OS 2.1, 3.2

- Decide if you want to use FAT16 or FAT32 for your file system. If you choose FAT16, you can later convert to FAT32 using the Windows Drive Converter. After Windows 98 is installed, to access the Converter, click **Start**, **Programs**, **Accessories**, **System Tools**, **Drive Converter (FAT32)**. Or you can use the Run dialog box: For the 16-bit version, enter **cvt.exe**, and for the 32-bit version, enter **cvt1.exe** and click **OK**. The Drive Converter wizard steps you through the process.

Performing the Installation

A+ OS 2.1

If you are performing a clean install on a freshly formatted hard drive, you must boot the PC from a floppy disk. Windows 9x comes on a set of floppy disks or on a CD. If you are installing the OS from floppy disks, boot from the Windows 9x Disk 1 and enter the **A: Setup.exe** command at the DOS prompt (where *A* is the letter of your floppy disk drive). The Windows 9x setup screen appears. Follow the directions on the screen. If you are installing the OS from a CD, after you have booted from a floppy disk, insert the CD in the CD-ROM drive and enter the command **D:\Setup.exe**, substituting the drive letter of your CD-ROM drive in the command line.

A+ OS 2.1

If this is an upgrade installation, do the following to get to the setup screen:

1. Start the PC, loading the current operating system.
2. Close all open applications including any antivirus software running.
3. Insert the CD in the CD-ROM drive or the floppy disk in the floppy drive. When upgrading from Windows 95 to Windows 98, open the Run dialog box and enter the command **D:\Setup.exe**, substituting the drive letter for the CD-ROM drive or floppy drive in the command line. Click **OK**. When upgrading from Windows 3.x to Windows 9x, enter the same command using File Manager.
4. Follow the instructions on the setup screen.

When installing Windows 9x, you are given the option of creating the Startup disk discussed in earlier chapters. Be sure to do that to help prepare for emergencies. Also during the installation, you will be asked to choose from four setup options:

- ***Typical*** This option installs all of the components that are usually installed with Windows 9x. Most often, this is the option to choose.
- ***Portable*** Use this option when installing Windows 9x on a notebook computer.
- ***Compact*** Use this option if you are short on hard drive space and want the smallest possible installation. No optional components are installed during the installation. After the installation, if you need a component, you can install it by double-clicking the Add/Remove Programs icon in the Control Panel.
- ***Custom*** Use this option if you know you will need components that are not normally installed under the Typical installation. You are given the opportunity to select any group of components to be included in the installation.

Windows 9x Setup begins installation in real mode and then later switches to protected mode. During real mode, it runs ScanDisk, checks for existing Windows software, performs several system checks, loads the extended memory driver, looks for existing TSRs, and starts Windows, if it is not already started. This Windows logo screen is the first thing the user normally sees during the installation. Setup then switches to protected mode.

Setup creates the registry, getting it ready to contain the hardware information, and then searches for hardware. It loads its own drivers for the detected hardware, or, if it cannot detect the hardware, requests the drivers from the user. The drivers are copied to the hard drive.

Up to this point, if Setup fails and you reboot the PC, you boot into DOS. Next Setup alters the boot records on the hard drive to point to the Windows 9x file, Io.sys, rather than to the DOS hidden files. Now, if Setup fails and you reboot, you reboot into Windows 9x.

During a normal installation, the PC reboots and Windows 9x is loaded. Some initial startup programs are run to set the time zone and to change existing application programs to Windows 9x. Depending on the hardware present, the PC may reboot again to load new drivers.

A+ OS 2.1

During the installation, Setup is recording information into log files. The primary log file is Setuplog.txt, a text file that is used by Windows to determine how far it got into the installation when it is recovering from a crash. The Detection Log (Detlog.txt) keeps a record of hardware detected. If the system fails to respond during the hardware detection phase, an entry is recorded in Detcrash.log, a binary file used by Windows to help recover from a crash caused by a problem with hardware. Windows does not use the contents of Detlog.txt; it is created only for the benefit of the user.

For example, if Setup suspects that a network card is present, because it sees a network driver installed in CONFIG.SYS, it records in Setuplog.txt and Detlog.txt that it is about to look for the card. If it successfully finds the card, it records the success in Detlog.txt. However, if an error occurs while Setup is searching for the card, an entry is made in the Detcrash.log file.

If the system crashed while trying to detect the network card and then Setup is restarted, it looks at Detcrash.log and Setuplog.txt to determine what it was trying to do at the time of the crash. It skips that step and goes on to the next step, so that it doesn't make the same mistake twice.

Even though Setup might crash several times during the installation process, progress is still being made. By reading the content of the log files, setup is able to skip those steps that caused a problem and move forward. Be careful not to delete the log files during the installation process, especially if you've just experienced a crash. Also, restart by using the power on/power off method so that the ISA bus is fully initialized, which does not always happen during a warm boot.

In certain situations you might want to force Setup to begin installation at the beginning instead of looking to Setuplog.txt for the entry point, for example when you think you might have resolved a problem with hardware and want Setup to attempt to find the hardware again. To do that, delete Setuplog.txt to force a full restart.

Customizing Setup

Windows and other software store information about an installation on the setup CD or floppy disks in information files. These **information files** are text files with an .inf file extension. One .inf file is Msbatch.inf. Information about your installation can be stored in this, which can then be used to do a hands-free installation. All the questions that a user must answer during an installation can be answered by entries in this file so that a user has little to do but begin the installation. For more information about hands-free installations, see the Microsoft Windows 95 or Windows 98 Resource Kit by Microsoft Press.

You can add several switches to the Setup.exe command that starts the setup process. Some of these switches and what they do are listed in Table 12-4.

Configuring the Windows 9x Startup with Msdos.sys

Recall that Msdos.sys plays an entirely different role in the boot process of Windows 9x than it does in the DOS boot process. In Windows 9x, this text file can contain several parameters that mainly affect how the OS boots. The file is a hidden, read-only, system file, so before

Table 12-4 Switches for the Setup command

Switch	Description
Setup /?	Display help for each command-line switch
Setup /D	Don't use the existing version of Windows to begin Setup. Use this option if you suspect corrupted Windows system files when upgrading Windows.
Setup /IC	Perform a clean boot. Use this option if you suspect drivers loaded from Autoexec.bat or Config.sys are causing a problem with the installation.
Setup /IH	Run ScanDisk in the foreground so that you can view results. Use this option if setup failed earlier and you want to check for hard drive corruption.
Setup /IL	Load the driver for a Logitech mouse. Use this option if you are using a Logitech Series C mouse.
Setup /IN	Do not set up the network.
Setup /IS	Do not run ScanDisk.
Setup /PI	Keep hardware settings that are not default settings. Use this option if a previous try of the installation caused a legacy hardware device to fail.

you can edit it, you must first use the ATTRIB command to make the file available for editing. Also, make a backup copy of the file in case you want to revert to the form it was in before the changes were made.

Follow these steps to change the options in Msdos.sys:

- Go to a DOS command prompt.
- Go to the root directory of your hard drive by typing:

  ```
  CD\
  ```

- Make the file available for editing by typing:

  ```
  ATTRIB -R -H -S MSDOS.SYS
  ```

- Make a backup copy of the file by typing:

  ```
  COPY MSDOS.SYS MSDOS.BK
  ```

- Use EDIT.COM to edit the file by typing:

  ```
  EDIT MSDOS.SYS
  ```

- Save the file and return it to a hidden, read-only, system file by typing:

  ```
  ATTRIB +R +H +S MSDOS.SYS
  ```

Table 12-5 lists each entry in the file and its purpose. You can refer back to this table as you read about the different options available when installing and configuring Windows 9x.

In Windows Explorer, by default, Windows 9x hides the file extensions of those files that it knows which application to use to open or execute the file. For example, just after the installation, it hides .exe, .com, .sys, and .txt file extensions, but will not hide .doc, .ppt, or .xls files until the software to open these files has been installed. To display all file extensions, click View, Folder Options and click the View tab. Unselect Hide file extensions for known file types.

Table 12-5 Contents of the Msdos.sys file options section

Command Line Variable Name	Purpose of the Values Assigned to the Variable
BootMulti	0 = (Default) Boot only to Windows 9x. 1= Allows for a dual boot
BootWin	1 = (Default) Boot to Windows 9x. 0 = Boot to previous version of DOS.
BootGUI	1 = (Default) Boot to Windows 9x with the graphic user interface. 0 = Boot only to the command prompt for DOS 7.0 or 7.1. (AUTOEXEC.BAT and CONFIG.SYS will be executed, and you will be in real-mode DOS.)
BootMenu	0 = (Default) Don't display the Startup Menu. 1= Display the Startup Menu.
BootMenuDefault	1 through 8 = The value selected from the Startup Menu by Default (Normally this value should be 1.)
BootMenuDelay	n = Number of seconds delay before the default value in the Startup Menu is automatically selected
BootKeys	1= (Default) The function keys work during the boot process (F4, F5, F6, F8, Shift+F5, Ctrl+F5, shift+F8). 0 = Disable the function keys during the boot process. (This option can be used to help secure a workstation.)
BootDelay	n = Number of seconds the boot process waits (when it displays the message "Starting Windows 95" or "Starting Windows 98") for the user to press F8 to get the Startup Menu (default is 2 seconds)
Logo	1= (Default) Display the Windows 9x logo screen. 0 = Leave the screen in text mode. (You can change the logo screen to be another .bmp file. The Windows 9x logo file is stored in Logo.sys in the root directory. Rename it, and name a new .bmp file to Logo.sys to customize this startup screen.)
Drvspace	1= (Default) Load Drvspace.bin, used for disk compression, if it is present. 0 = Don't load Drvspace.bin.
DoubleBuffer	1= (Default) When you have a SCSI drive, enables double buffering for the drive (see the drive documentation) 0 = Don't use double buffering for the SCSI drive.
Network	1= If network components are installed, include the option, "Safe mode with network support" in the Startup Menu. 0 = Don't include the option on the Startup Menu. (This will normally be set to 0 if the PC has no network components installed. The Startup Menu will be renumbered from this point forward in the menu.)
BootFailSafe	1= (Default) Include Safe mode in the Startup Menu. 0 = Don't include Safe mode in the Startup Menu.

Table 12-5 Contents of the Msdos.sys file options section (continued)

Command Line Variable Name	Purpose of the Values Assigned to the Variable
BootWarn	1= (Default) Display the warning message when Windows 9x boots into Safe mode. 0 = Don't display the warning message.
LoadTop	1= (Default) Load COMMAND.COM at the top of conventional memory. 0 = Don't load COMMAND.COM at the top of conventional memory. (Use this option when there is a memory conflict with this area of memory.)

Figure 12-11 shows a sample Msdos.sys file. The lines containing x's at the bottom of the file are used to ensure that the file size is compatible with other programs.

```
[Paths]
WinDir=C:\WIN95
WinBootDir=C:\WIN95
HostWinBootDrv=C

[Options]
BootMulti=1
BootGUI=1
BootMenu=1
Network=0
;
;The following lines are required for compatibility with other programs.
;Do not remove them (MSDOS.SYS needs to be >1024 bytes).
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxa
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxb
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxc
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxd
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxe
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxf
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxg
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxh
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxi
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxj
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxk
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxl
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxm
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxn
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxo
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxp
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxq
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxr
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxs
```

Figure 12-11 A sample Msdos.sys file

A+ OS 2.1, 2.2

Installing Windows 9x over DOS and Windows 3.x

If DOS and Windows 3.x reside on the PC prior to installing Windows 9x, you have the choice of installing Windows 9x over DOS and Windows 3.x or installing Windows 9x in a separate directory from Windows 3.x so that you can still run 16-bit programs under Windows 3.x.

The advantages of overwriting Windows 3.x and DOS are:

- Less hard drive space is used.
- Windows 9x Setup copies information about existing application programs from the .ini files of Windows 3.x into the Windows 9x registry, eliminating the need for you to install the existing programs into Windows 9x.
- These programs are added to the Start menu of Windows 9x.
- Existing programs can find their .dlls in the same Windows\System folder as they did with Windows 3.x.

The advantages of installing Windows 9x in a separate directory are:

- You can create a **dual boot** and run the PC with either Windows 3.x or Windows 9x.
- The Windows\System folder of Windows 9x is not cluttered with old, outdated .dlls that current programs no longer use. In effect, you get a "fresh start."

If you install Windows 9x in a separate directory, you must then reinstall each application program that you want to run under Windows 9x, so that it gets information into the Windows 9x registry, its programs are listed under the Start menu, and its .dlls and other supporting files are copied to the Windows\System folder.

Dual Boot Between Windows 9x and DOS with Windows 3.x

You can install DOS with Windows 3.1 and Windows 9x on the same hard drive, so that you can use software made for each operating environment within its native OS. See Chapter 7 for details on how to do this.

Plug and Play and Hardware Installations

Plug and Play (PnP) is a set of design specifications for both hardware and software that work toward effortless hardware installations. For a system to be truly Plug and Play, it must meet these criteria:

- The system BIOS must be PnP.
- All hardware devices and expansion cards must be PnP-compliant.
- The OS must be Windows 9x or another OS that supports PnP.
- A 32-bit device driver (VxD) must be available (provided by the device manufacturer or Windows).

If all these things are true, hardware installation should be very easy and is only a matter of installing the new hardware device, turning on the PC, and perhaps providing the 32-bit driver, if it is not included with Windows 9x. During the boot process, Windows 9x surveys the devices and their needs for resources and allocates resources to each device. Windows 9x is free to assign these resources to the devices and avoids assigning the same resource to two devices. For PnP to work, each device in the system must be able to use whatever resources the OS assigns to it.

How Plug and Play Works

A+ OS 2.4

A Plug and Play OS like Windows 9x provides two main services: resource management and run-time configuration. **Resource management** occurs at startup as system resources are allocated to devices. **Run-time configuration** is an ongoing process that monitors any changes in system devices, such as the removal of a PC Card on a notebook computer or docking and undocking a notebook computer to and from a docking station. The BIOS must be able to recognize these changes during OS run time (any time the OS is running) and communicate them to the OS.

Windows 9x uses four components in implementing PnP architecture:

- The **configuration manager** controls the configuration process of all devices and communicates these configurations to the devices.
- The **hardware tree** is a database built each time Windows 9x starts up that contains a list of installed components and the resources they use.
- The **bus enumerator** locates all devices on a particular bus and inventories the resource requirements for these devices.
- The **resource arbitrator** decides which resources get assigned to which devices.

When Windows 9x is started, if the system board BIOS is PnP, the configuration manager starts the PnP process by receiving the list of devices from the BIOS. If the BIOS is not PnP, the bus enumerator for each bus on the system provides the information to the configuration manager. The configuration manager oversees the process of assigning resources by loading one device driver after another for each installed device, and instructing the driver to wait until resources have been assigned to it.

The configuration manager performs a process of examining and reexamining required resources until it determines an acceptable configuration of all resources and devices. For example, for ISA devices, according to the PnP standards, each device has a unique 72-bit ID derived from the manufacturer ID, a product ID, and a serial number. Each ISA device competes for resources, and the device with the largest-value ID is assigned resources first. The bus enumerator manages this process and receives the resource assignments from the configuration manager. The configuration manager interacts with the resource arbitrator, allowing the arbitrator to determine what resources are assigned, and then receives that information from the arbitrator, passing it on to the bus enumerator.

The bus enumerators collectively build the hardware tree, which is stored in memory. Information to build the hardware tree comes from the configuration at the current moment as well as from information kept in the registry about devices that have been installed, including what device drivers are used to operate the device and user-defined settings for the device. The hardware tree is built each time Windows 9x is started, and is dynamically changed as hardware is plugged and unplugged while the system is running.

Plug and Play BIOS

A+ OS 2.4

As discussed in previous chapters, BIOS that is PnP-compliant gathers resource configuration information prior to loading Windows 9x, presenting to Windows 9x details it can use to complete the process. System boards manufactured after 1994 most likely contain PnP BIOS. PnP BIOS can also be ESCD (extended system configuration data) BIOS. ESCD BIOS creates a list of configuration changes that you have made manually when installing legacy devices and stores that list on the BIOS chip. Even if the hard drive crashes or you must reload Windows 9x, the configuration changes are still available from the BIOS when it goes through the boot process and presents the information to Windows 9x at startup.

To know if your BIOS is PnP, look for a message about the BIOS type on the startup screen. Information about the BIOS might also be displayed on the CMOS setup screen or written on the BIOS chip (see Figure 12-12). Use MSD and choose Computer from the menu to get information about your BIOS. The documentation for the system board should also say whether or not the BIOS is PnP.

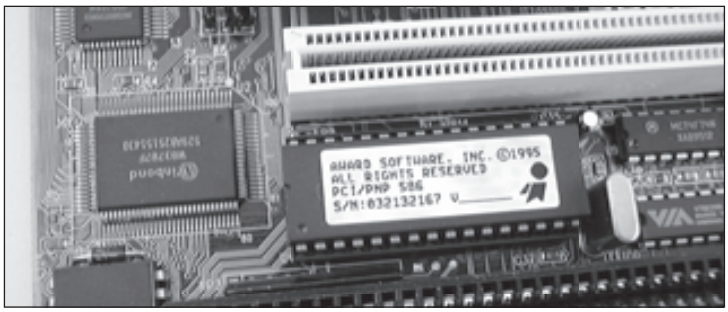


Figure 12-12 Plug and Play BIOS is found on most system boards built after 1994

TIP

If the BIOS is not Plug and Play, you can still use Plug and Play Windows 9x software for hardware devices that are Plug and Play. However, you might need to manually configure the hardware, or, in some cases, disable the Plug and Play features of the interface card.

Installing New Hardware

A+ OS 2.4

Windows 9x provides better support during the installation of new hardware devices than does Windows 3.x. If the computer system is completely PnP-compliant, installations are automated and go very smoothly. However, with older hardware devices and older drivers, problems can occur that must be resolved manually.

This section looks at three sources of problems with hardware installations and what to do about them. Table 12-6 summarizes these problems and their solutions. A device or expansion card that is not PnP is a **legacy** (handed down from the past) device. Legacy devices are not able to have their resources assigned to them by PnP. If the device driver is an older 16-bit driver, there might be a problem with Windows 9x installing and using the driver. Also, Windows 9x might or might not have a built-in driver for the device. Each problem and how to address it is discussed below.

Table 12-6 Hardware device installation problems and solution

Source of the Problem	Nature of the Problem	Solution to the Problem
Unsupported devices	Windows 9x does not have a built-in device driver designed for the device	Provide a device driver from the manufacturer or use a substitute
16-bit drivers	Windows 9x has trouble initially recognizing and installing the driver	First, install the driver using DOS or Windows 3.x
Legacy cards	Can cause a conflict of resources (IRQ, I/O addresses, upper memory addresses) between two devices	Change the DIP switches or jumpers on the card to use different resources

Unsupported Devices

New hardware devices are installed from the Control Panel using the Add New Hardware icon. Windows 9x uses the Add New Hardware Wizard to recognize the device, install the correct driver, and allocate the correct resources to the device. The following are general directions to use if Windows 9x does not recognize the device you are installing. After physically plugging in or installing the device, turn on the PC, go to the Control Panel, and choose **Add New Hardware**. When you see the options in Figure 12-13, answer **Yes** to allow Windows to search for the device, and then click **Next**.

If Windows 9x locates a new device and recognizes it to be PnP, as shown in Figure 12-14, simply click **Next** and the work is done.

A+ OS 2.4

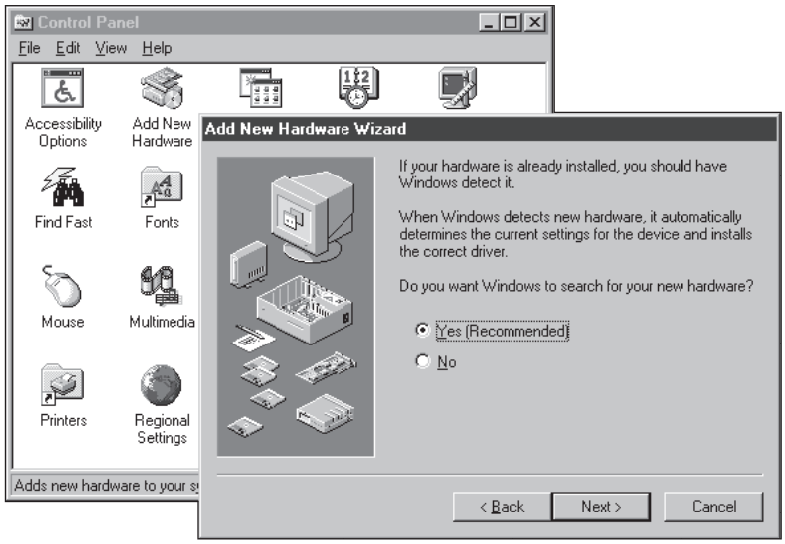


Figure 12-13 Add New Hardware Wizard

Figure 12-14 If Windows 9x recognizes the new device it can complete the installation with no help

If Windows 9x detects the device and recognizes that it is not PnP, it suggests resources that can be assigned to the device, as seen in Figure 12-15. It's left up to you to manually configure the device to use the suggested resources or to select other resources.

A+ OS 2.4

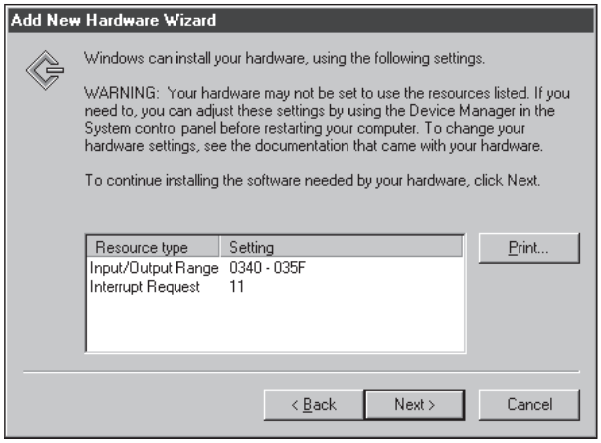


Figure 12-15 If the new device is not PnP, Windows 9x suggests the resources for a legacy device to use

However, if Windows 9x cannot detect a new device, it will ask for your help, as shown in Figure 12-16. When you click **Next**, a list of devices is displayed, as shown in Figure 12-17. Select the device and click **Next**. The list of supported manufacturers and models for that device type is displayed. For example, Figure 12-18 shows a list of CD-ROM drive controllers. You can select a manufacturer and model from the list to use a Windows driver, or you can click **Have Disk** and supply the driver from a device manufacturer's disk or CD-ROM. If you don't have the manufacturer's driver and the device is not listed, try obtaining the driver from the manufacturer's Web site.

When you have a choice between installing a Windows 9x device driver or a driver supplied by the device manufacturer, you will usually get better results using the manufacturer's driver.

Figure 12-16 The new device is not detected or recognized by Windows 9x

A+ OS 2.4

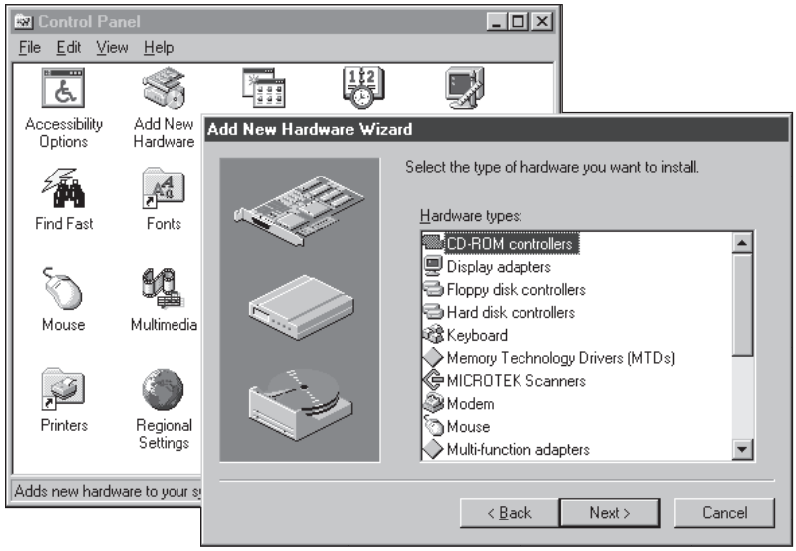


Figure 12-17 When Windows 9x cannot recognize the new hardware, you must select the device

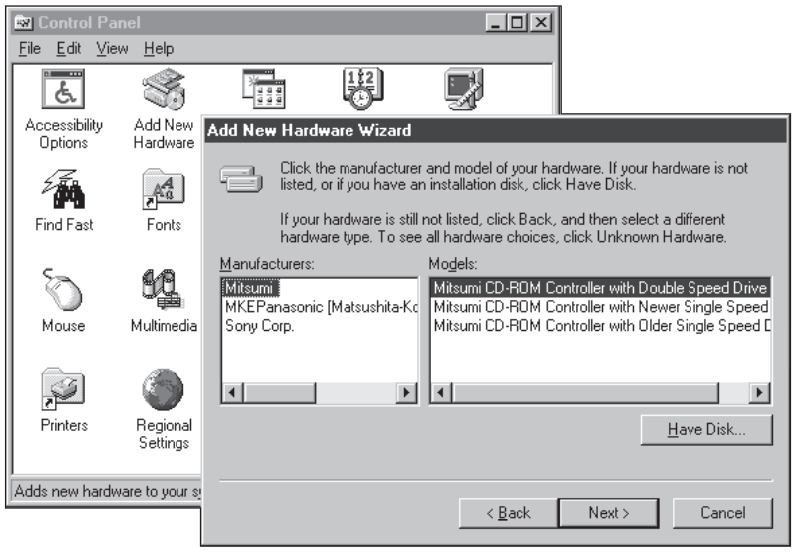


Figure 12-18 List of manufacturers and models of supported CD-ROM drive controllers

Problems with 16-Bit Drivers

A+ OS 3.2

When you supply a device driver to Windows 9x for an unsupported device, Windows 9x copies one or more device driver files to the hard drive and make the appropriate entries in the registry so that it will load the driver each time it starts up. If the driver is a 16-bit

A+ OS 3.2

driver, it also makes the appropriate entries in the AUTOEXEC.BAT and CONFIG.SYS files. If Windows 9x has problems locating and installing a 16-bit driver, then you have three choices. You can contact the manufacturer of the device for an updated driver, you can ask them to recommend a substitute driver, or you can install the driver using DOS or Windows 3.x. If the device driver has an install program, run the install program under either DOS or Windows 3.x, depending on what the install program requires. It makes the correct entries in the AUTOEXEC.BAT and CONFIG.SYS files and copies the driver files to the hard drive. You should then be able to install the device in Windows 9x with no problems.

Problems with Legacy Cards

When you are installing Windows 9x onto a PC that has previously worked well with DOS and Windows 3.x, legacy cards will not normally be a problem because the conflict of resources will usually have already been resolved when the cards were first installed.

A problem arises if the device is a legacy device and requests resources that conflict with another legacy device already installed. You then must intervene and change the jumpers or DIP switches on one of the devices to force it to use a different resource.

Using Device Manager for Troubleshooting

During the installation, Windows 9x might inform you that there is a resource conflict, or the device might simply not work. Use Device Manager as a useful fact-finding tool for the resolution of the problem. Open the **Control Panel**, choose **System**, and select the **Device Manager** tab. The list of devices is displayed, as seen in Figure 12-19. A + beside the device name indicates that you can click the device for a list of manufacturers and models installed. The open diamond symbol indicates a SCSI device. Symbols that indicate a device's status are:

- A red X through the device name indicates a disabled device.
- A yellow exclamation point indicates a problem with the device.
- A blue I on a white field indicates automatic settings were not used, and resources have been manually assigned. It does not indicate a problem with the device.

To see a better explanation of a problem, click the device and select **Properties**. The Device Properties dialog box opens, which can give you helpful information about solving a problem including I/O addresses, DMA channels, and IRQs used by the device as well as the names of devices that are also attempting to use the same resources.

In fact, before you start the hardware installation, you might want to use Device Manager to print a summary of all hardware installed on the PC and resources being used. This printout can be a record of your starting point before the installation as well as a tool to help resolve conflicts during the installation. To print this summary, access Device Manager and click **Print**. From the Print dialog box, select **All Devices and System Summary** for a complete listing.

A+ OS 3.2

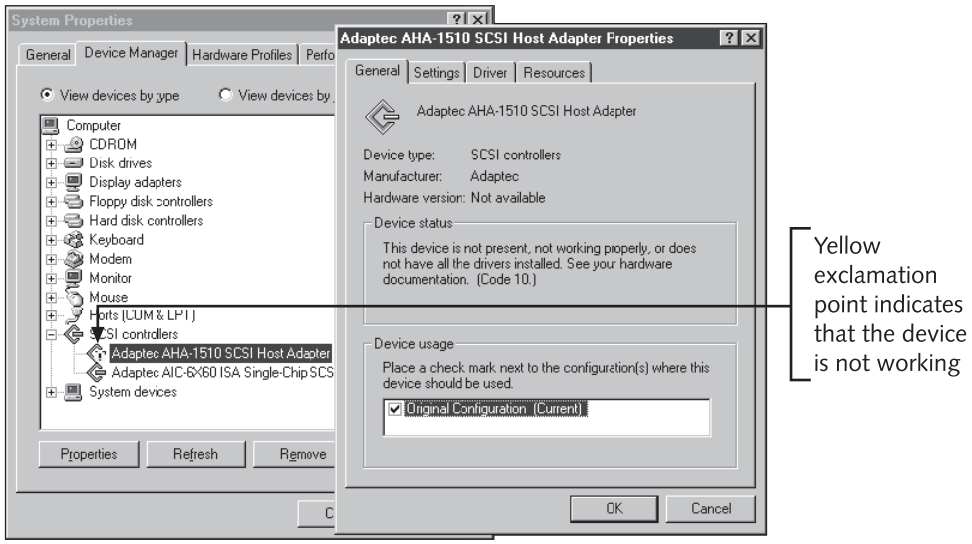


Figure 12-19 The Properties box of an installed device that is not working

Troubleshooting with Windows Help

For more information from Windows 9x on how to resolve a problem, click **Start**, click **Help**, and click **Troubleshooting**. The Help information includes suggestions that can lead you to a solution. For example, in Figure 12-20, the Hardware Troubleshooter suggests that you check to see that the device is not listed twice in Device Manager. If this were the case, you would have to remove the second occurrence of the device.

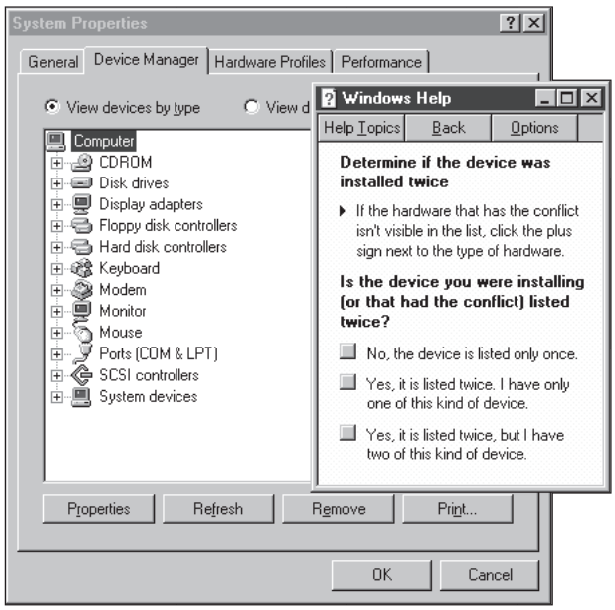


Figure 12-20 Troubleshooter making a suggestion to resolve a hardware conflict

The Windows 9x Registry

The Windows 9x registry is intended to replace the many .ini files that Windows and Windows applications software used under Windows 3.x. Organizing the information contained in these older .ini files is accomplished by using a hierarchical database with a tree-like, top-to-bottom design. All kinds of information is stored in the registry, including system configurations, user settings, Device Manager information, applications software settings, hardware settings, and so on. In this section you will examine how the registry is organized, what kinds of information are in the registry, how and why you might edit the registry, and how to recover from a corrupted registry.

How the Registry Is Organized

A+ OS 1.1

Recall that the Windows 9x System.ini file contains setup parameters. Refer back to Figure 12-6, which shows a portion of the System.ini file. Notice that section names appear in square brackets, key names to the left of the equal signs, and values assigned to these key names to the right of the equal signs. The Windows 9x registry takes on a similar design but enhances it by allowing for keys to cascade to several levels on the tree. Figure 12-21 shows a portion of a Windows 9x registry. Consider names on the left of the window as similar to section names in System.ini; these names are called **keys** by Windows 9x. On the right of the window are value names, such as ScreenSaveTime, and to the right of each name is the **value data** assigned to that name, such as "60." The value names, called values by Windows 9x, are similar to the key names in System.ini, and the value data are similar to the values assigned to key names in System.ini.

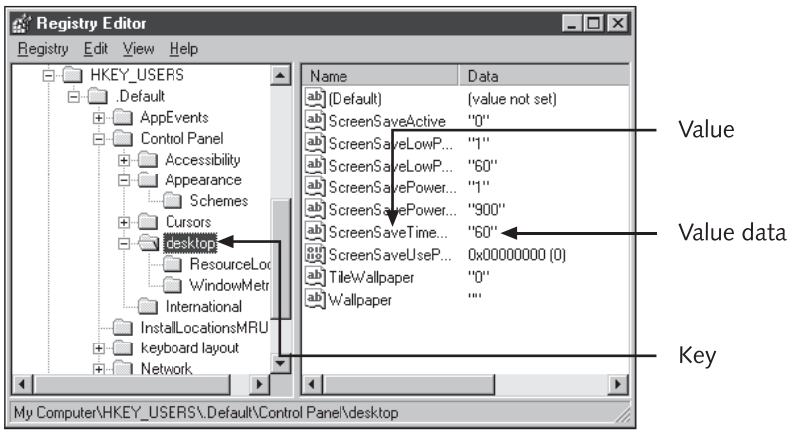


Figure 12-21 Structure of the Windows 9x registry

The registry is organized into six major keys or branches of the registry tree, which are listed in Table 12-7. The registry is contained in two files, System.dat and User.dat, located in the Windows directory as hidden, read-only, system files, although the information forms only a single database.

In a network environment, Windows 9x can use policy files to define user, network, and computer settings. A policy file has a .pol file extension; examples are Policy.pol and Config.pol. Using the System Policy Editor, a network administrator can create a policy file that affects the entire network, a group of users, or a single user on the network. The policy file can reside on the network server and is read when Windows 9x first boots. Entries in the policy file override entries in the Windows 9x registry.

Table 12-7 Six major branches, or keys, of the Windows 9x registry

Key	Description
HKEY_CLASSES_ROOT	Contains information about file associations and OLE data (This branch of the tree is a mirror of HKEY_LOCAL_MACHINE\Software\Classes.)
HKEY_USERS	Includes user preferences, including desktop configuration and network connections
HKEY_CURRENT_USER	If there is only one user of the system, this is a duplicate of HKEY_USERS, but for a multiuser system, this key contains information about the current user preferences.
HKEY_LOCAL_MACHINE	Contains information about hardware and installed software
HKEY_CURRENT_CONFIG	Contains the same information in HKEY_LOCAL_MACHINE\Config and has information about printers and display fonts
HKEY_DYN_DATA	Keeps information about Windows performance and Plug and Play information

Recovering from a Corrupted Registry

Windows 9x maintains a backup copy of the two registry files called System.da0 and User.da0. Each time Windows boots successfully, it makes a backup copy of these two files. If Windows 9x has trouble loading and must start in safe mode, it does not back up the registry files.

If Windows 9x does not find a System.dat when it starts, it automatically replaces it with the backup System.da0. If both System.dat and User.dat are missing, or if the "WinDir=" command is missing in Msdos.sys, Windows 9x tells you that the registry files are missing and starts in safe mode. It then displays the Registry Problem dialog box. Click the **Restore From Backup** and **Restart** buttons. The registry files are restored from System.da0 and User.da0. If these files are also missing, the registry cannot easily be restored. You can either restore the files from your own backups or run Windows 9x Setup. There is another option. Look for the file System.1st in the root directory of the hard drive. This is the System.dat file created when Windows 9x was first installed. In an emergency, you can revert to this file.

A+ OS 3.2

Windows 98 Registry Checker Windows 98 offers a utility that is not available with Windows 95 called the Registry Checker. It automatically backs up the registry each day, and by default, it keeps the last five days of backups. In an emergency, you can recover the registry from one of these backups. You can also tell Registry Checker to make an additional

A+ OS 3.2

back up on demand, such as when you have just made changes to the registry and want to backup these changes before you make new changes.

To access Registry Checker, select **Start**, point to **Programs**, **Accessories**, **System Tools**, and then click **System Information**. The Microsoft System Information window opens (see Figure 12-22). From the menu bar, select **Tools** and then **Registry Checker**. Registry Checker tells you if the registry is corrupted and will fix it, if allowed. You can also create a new backup at this time.

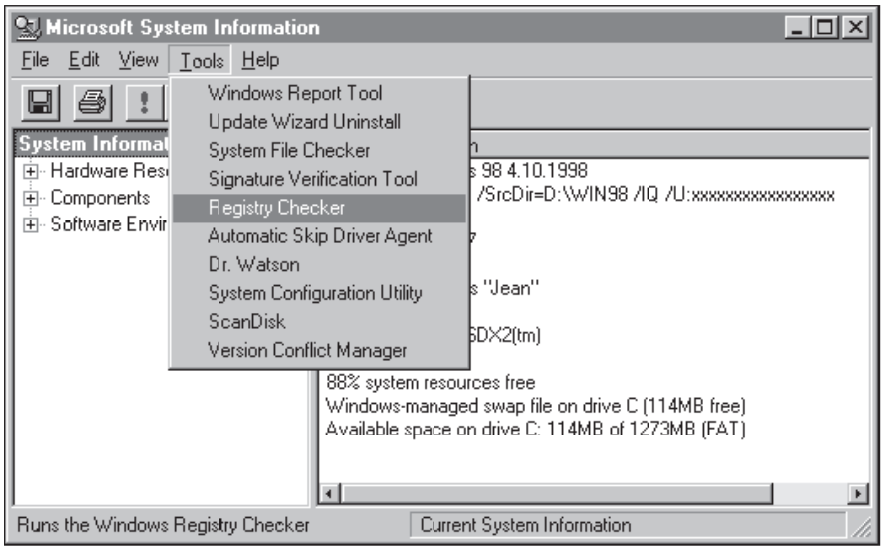


Figure 12-22 The Registry Checker is available under Programs, Accessories, System Tools, System Information Tool; it is used to back up, restore, and repair the Windows 98 registry

Backups are kept in cabinet files in the \Windows\Sysbckup folder as rb001.cab, rb002.cab, and so on. To revert to one of these backups, you must first be in real-mode DOS. Either boot from a bootable disk or boot to a DOS prompt from the Windows 98 startup menu. From the DOS prompt (not a DOS box within a Windows session), use these commands to repair or recover the registry:

- ScanReg/Restore — Restores the registry from a previous backup. A screen is displayed asking you which backup to use.
- ScanReg/Fix — Repairs the corrupted registry. If the problem is inherent to the registry itself, this might work. If the problem is that you want to undo a successful change to the registry, then use the Recover option instead.
- ScanReg/Backup — Creates a new backup of the registry at the DOS prompt. Don't do this if the registry is giving you problems.

A+ OS 3.2

- ScanReg/Opt — Optimizes the registry. ScanReg will look for and delete information in the registry that is no longer used. This reduces the size of the registry, which might speed up booting.
- ScanReg/? — Help feature of ScanReg

Nuts & Bolts Registry Wizard Nuts & Bolts Registry Wizard can back up and restore the registry, clean the registry of unneeded data, search for and repair registry orphans (registry entries that refer to files that have moved or no longer exist), and tune up and optimize the registry for better performance. Click **Start**, point to **Programs**, click **Nuts & Bolts**, and select **Registry Wizard** from the list of Nuts & Bolts utilities. Figure 12-23 shows the opening screen of the Registry Wizard. The best time to use the Registry Wizard is before the problem occurs by making a backup of the registry before installing new software or hardware, making significant system configuration changes, and before editing the registry.

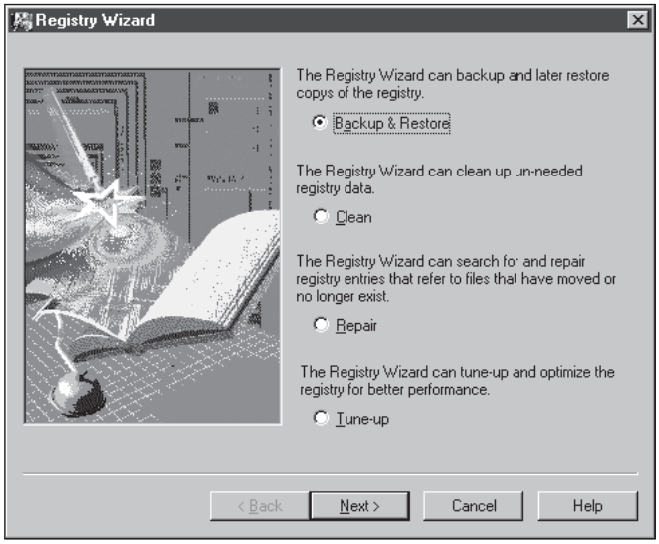


Figure 12-23 Nuts and Bolts Registry Wizard can back up, restore, clean, repair, and optimize the Windows 9x Registry

Modifying the Registry

When you make a change in Control Panel or Device Manager or many other places in Windows 9x, such as happens when you install software, the registry is modified automatically. For most users, this is the only way they will ever change the registry.

However, there are rare occasions when you might need to edit the registry manually. One example is when you have accidentally deleted the device driver for a hardware device, but Device Manager says that the device is still installed. Another example is when the wrong software starts when Windows 9x is loaded and you cannot correct the problem by changing the Startup folder. Both these problems can be corrected by manually editing the registry.

Editing the Registry

The first step in editing the registry is to back up the two files, System.dat and User.dat. Sometimes the files are small enough to fit on floppy disks and can be copied using Explorer. If the files are too large to copy to floppy disk, copy them to a different folder on the hard drive or use compression software such as PKZIP to copy them to floppy disks. For Windows 98, use Registry Checker to back up the registry. Utility software, including Nuts & Bolts and Norton Utilities, has a registry editor that allows for backing up the registry before entering the editor. The following directions use Windows 9x Regedit to edit the registry.

A+ OS 1.1, 3.2

After backing up the registry files, the next step is to use Regedit.exe, located in the Windows folder. You can use Explorer to locate the file, then double-click it, or click **Start**, and then **Run**, and type **Regedit** in the Run dialog box. When you do, the dialog box in Figure 12-24 opens. Open one branch of the tree by clicking on the + sign to the left of the key, and close the branch by clicking on the – sign.

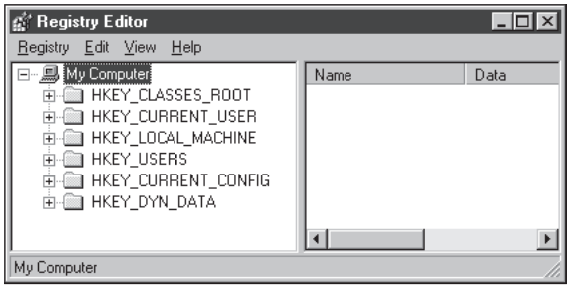


Figure 12-24 The six major keys, or branches, of the registry seen in the Registry Editor

12

To search for an entry in the registry, click **Edit**, and **Find**. The Find dialog box is displayed, as in the screen in Figure 12-25, which is ready to find the text "software" in the registry. Enter the key, the value, or the value data, and click **Find Next**. You can choose to search keys, values, and/or value data by clicking on the check boxes in the dialog box.

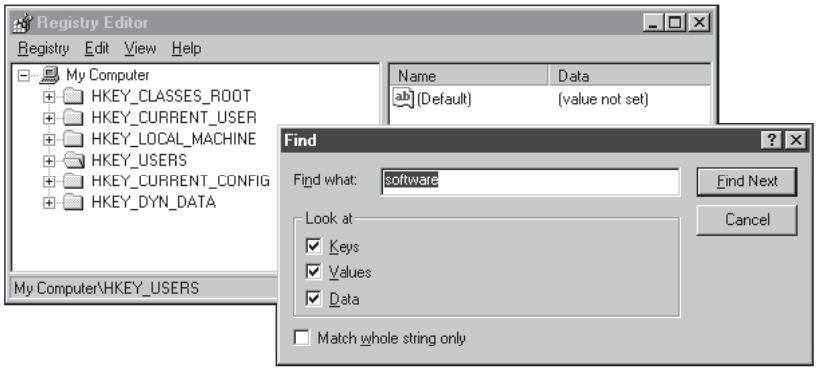


Figure 12-25 Searching for entries in the registry

For example, suppose the wrong programs start when you load Windows 9x. First try to correct the problem without editing the registry. Using Explorer, open the Windows folder and then double-click the **Start Menu** folder. Delete any items that you don't want to start when you load Windows. If this does not correct the problem, the problem might be caused by a wrong entry recorded in the registry. Try editing the registry.

First locate the Shell Folders, which will be in the following branch:
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

Search for these keys and subkeys one at a time. (Search for HKEY_CURRENT_USER. After you have located it, search for Software, and continue through the list until you come to Shell Folders.)

The value name "Startup=" in the Shell Folders subkey should be "C:\Windows\Start Menu\Programs\Startup." If the data is incorrect, right-click **Startup** and select **Modify** from the shortcut menu. Figure 12-26 shows the Editing dialog box that is displayed. Change the value data and click **OK**.

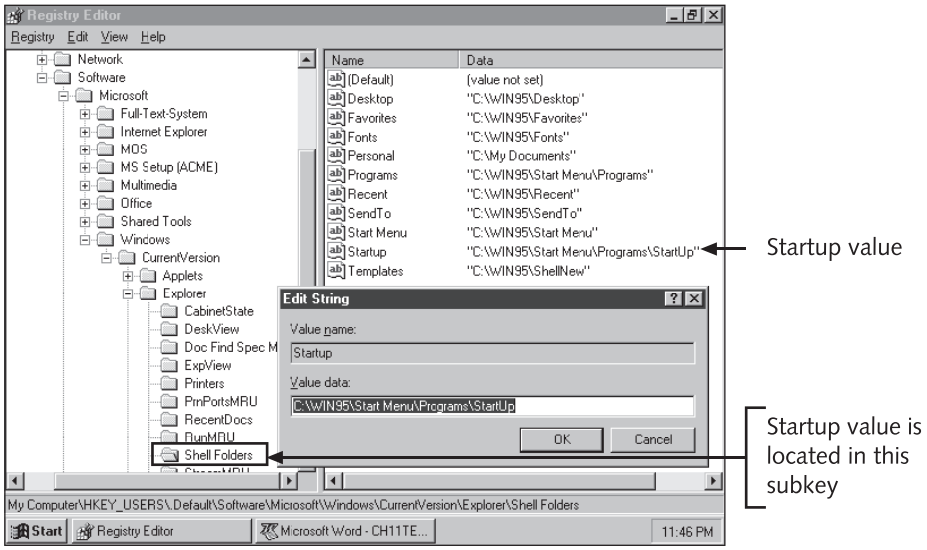


Figure 12-26 Editing an entry in the registry using Regedit.exe

Tracking Changes to the Registry During Software Installation

Two utilities that can track changes not only to the Windows 9x registry but also to .ini files and folders are a shareware program, In Control 4, and Norton Utilities Registry Tracker. To use Norton Utilities Registry Tracker, click **Start**, **Programs**, **Norton Utilities**, **Norton Registry Tracker**, as shown in Figure 12-27. Figure 12-28 shows the Norton Registry Tracker main windows. When the utility first loads, it takes an Activation Snapshot of the system, which it will later use as the baseline to determine if future snapshots have detected any changes.

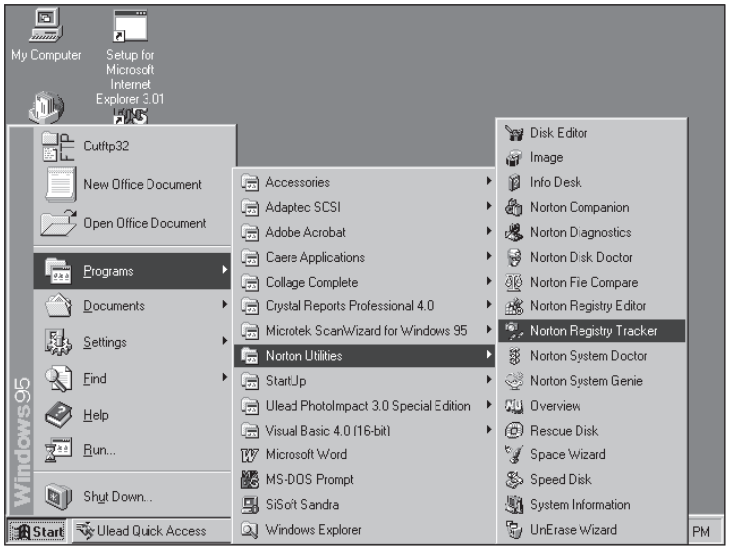


Figure 12-27 Finding Norton Utilities Registry Tracker

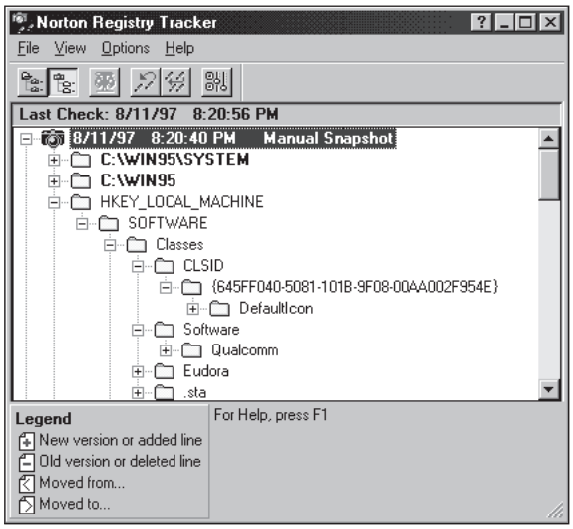


Figure 12-28 Norton Utilities Registry Tracker tracks changes to the Windows 9x registry

With the Norton Registry Tracker, you can first choose what it is you want to track and then take a snapshot of the system. Next install the software and then take another snapshot. Compare the two snapshots to see what the installation did to those things you chose to track.

To change the items tracked, click **Options**, and then **Settings** from the Norton Registry Tracker menu. The Tracker Settings screen is displayed, as in Figure 12-29. Select the tab **Registry Keys** to see a list of keys to be tracked. Norton Registry Tracker tracks the two keys and their subkeys, HKEY_LOCAL_MACHINE and HKEY_USERS, because these

keys contain the first copy of changes made when software is installed and user settings are changed. In Figure 12-30, the camera and the check mark beside the SOFTWARE subkey indicate that changes to it are to be tracked. If you want to track other keys, select the key and click **Track Key** or **Track Subkey**. On the Folders tab you can choose which folders to track. By default, the Windows folder and the Windows\System folder are tracked. Under text files, AUTOEXEC.BAT, CONFIG.SYS, and the .ini files are tracked. Return to the main window by clicking **OK**.

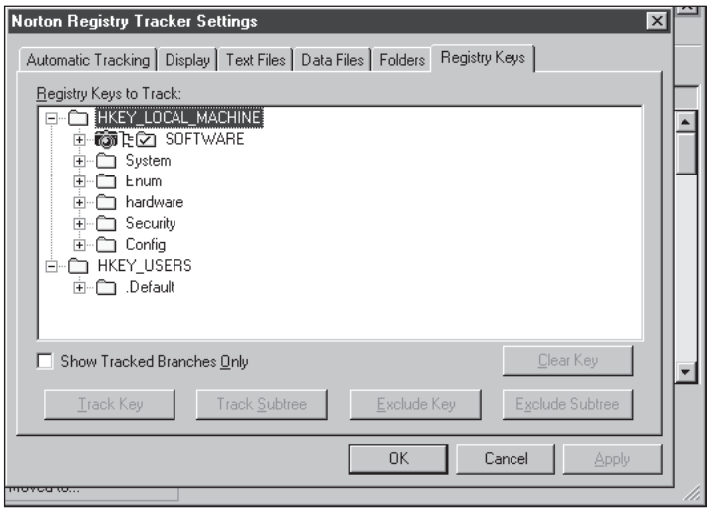


Figure 12-29 Select what you want Norton Registry Tracker to track during an installation

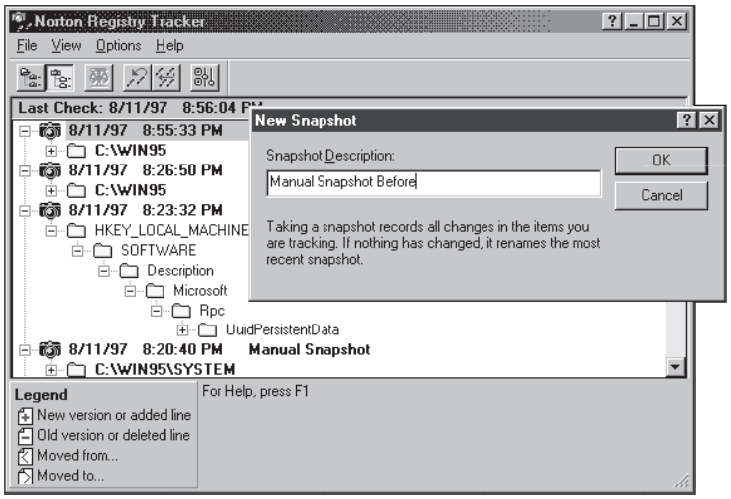


Figure 12-30 Naming the new snapshot

To take a snapshot, select **File**, **New Snapshot**. You see the dialog box in Figure 12-30, asking you to name the snapshot. Enter the name and click **OK**. The snapshot is made and compares the things being tracked to the original activation snapshot made when the utility was first loaded. In the snapshot, changed items are color-coded.

Take a snapshot before the installation, perform the installation, and take another snapshot after the installation. The Norton Registry Tracker can then provide a side-by-side view of the two snapshots to make comparisons easy. Always using Norton Registry Tracker in this way is a good practice whenever installing software.

SUPPORTING APPLICATIONS SOFTWARE WITH WINDOWS 9X

In this section we look at installing applications software, the special needs of DOS applications and problems that can arise when uninstalling software.

Installing Applications Software

A+ OS 2.4

If the software is 32-bit applications software designed to be used with Windows 9x, installation is easily done by using the Add/Remove Programs icon in Control Panel. For older 16-bit software, use the Run dialog box. Before you install any software, do the following:

Check Available Resources Check your computer resources to make sure you have (1) enough space on your hard drive, (2) the minimum requirements for memory, and (3) the proper CPU and video monitor—and that you can fulfill any other requirement of the particular software program. The minimum requirements for the software should be listed in the installation manual. Remember that you should not completely fill the hard drive with software and data.

For best performance with Windows 9x, allow a minimum of 100 MB of unused hard drive space for working temporary files used by applications.

Protect the Original Software For floppy disks, write-protect the original disks before you begin the installation. After the installation is complete, put the original disks or CD-ROM in a safe place.

A+ OS 2.4, 3.2

Back up the Registry and System Configuration Files Many older software packages will want to edit CONFIG.SYS, AUTOEXEC.BAT, Win.sys, and/or System.ini files during the installation. Newer software might add its own entries to the Windows registry. Before you begin the installation, make backup copies of all of these files so that you can backtrack if you want to.

If you are having difficulty with the installation, look at the Readme.htm hypertext file in the \Windows directory, which will point you to the Programs.txt file, also in the \Windows directory. If there is a problem with the software that was known when Windows was

A+ OS 2.4, 3.2

shipped, information about the problem and what to do about it might be in these text files. You can also check the web site of the software manufacturer or the Microsoft web site for additional insight.

A+ OS 2.4

A hypertext file is a text file that contains hypertext tags to format the file and create hyper links to different points in the file or to other files. Hypertext files are used on the World Wide Web and are read and displayed using a web browser such as Microsoft Internet Explorer or Netscape Navigator. To read a hypertext file using Windows Explorer, double-click the filename and your default browser opens the file.

Install the Software For software designed for Windows 9x, access the Control Panel and double click the Add/Remove Programs icon. Insert the CD in the CD-ROM drive or the floppy disk in the floppy disk drive and click the Install button. Follow directions on the setup screen. Or, for older software, click Start and Run, which displays the Run dialog box. Enter the drive and name of the installation program, for example, A:Install or D:Setup. Either way, the installation program loads and begins executing. If the installation program asks you a question you cannot answer, you can always abandon the installation and try again later.

Most software will ask you for a serial number unique to your copy of the software, which will probably be written on the CD-ROM or on the first floppy disk, or might be stamped on the documentation. Write the serial number on the floppy disk or on the CD case, so that you will still have it if the documentation is later lost. Copyright agreements often allow you to install the software on only one computer at a time. This serial number identifies the copy of the software that you have installed on this machine.

After the installation is complete and the software is working, update your backup copies of AUTOEXEC.BAT, CONFIG.SYS, System.ini, Win.ini, and the registry so that they, too, reflect the changes that the applications software made to these configuration files.

A software installation sometimes leaves files and folders in the Windows temporary directories. To conserve space on the hard drive, delete all files and folders under \Windows\Temp.

If an application locks up when you first open it, try deleting all files and folders under \Windows\Temp.

A+ OS 3.1

A troubleshooting tool you can use when you are having problems running an application is Dr. Watson. **Dr. Watson** is a Windows utility that can record detailed information about the system, errors that occur, and the programs that caused them in a log file named \Windows\Drwatson\WatsonXX.wlg, where XX is an incrementing number. Start Dr. Watson and then reproduce the application error. Then look at the events logged in the Dr. Watson window under the Diagnosis tab. Use this information to check the Microsoft web site, support.microsoft.com, for the problem and solution. For errors that you cannot reproduce at will, you can load Dr. Watson each time Windows starts by creating a shortcut to Drwatson.exe in the Startup folder.

Three ways to cause an application to automatically load at startup are to place a shortcut in the Startup folder, put the name of the program file in the Load= or Run= line in Win.ini or manually edit the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.

A+ OS 3.2

Problems with Software Conflicts

Suppose you have Microsoft Word installed correctly on your PC; it has worked well for some time. After you install several new applications, however, Word begins to give you problems. Icons that once were displayed correctly are now displayed as black objects, or, whenever you exit Word, Windows displays a General Protection Fault error message. You attempt to correct the problem by reinstalling Word, but the problem does not go away. What happened to Word, when you installed the other software, that cannot be repaired by reinstalling Word?

The answer is found in the directory \Windows\System. Take a look at this directory, as shown in Figure 12-31. The directory holds files that are used by Windows and other applications software programs. Several programs can use the same files found in this directory. The most common type of file found here has a **.dll** extension, which stands for **dynamic-link library**. These library files contain various programming routines that are used by many application programs to perform such common tasks as opening a database file or displaying a dialog box on the screen. Windows comes with many of these .dll files already installed, but some applications place other .dlls in the \Windows\System directory or replace existing copies of .dll files with more recent versions during installation. Most installations don't save previous copies of .dlls they update; the new application simply overwrites the file with its own version, without informing you of what it's doing.

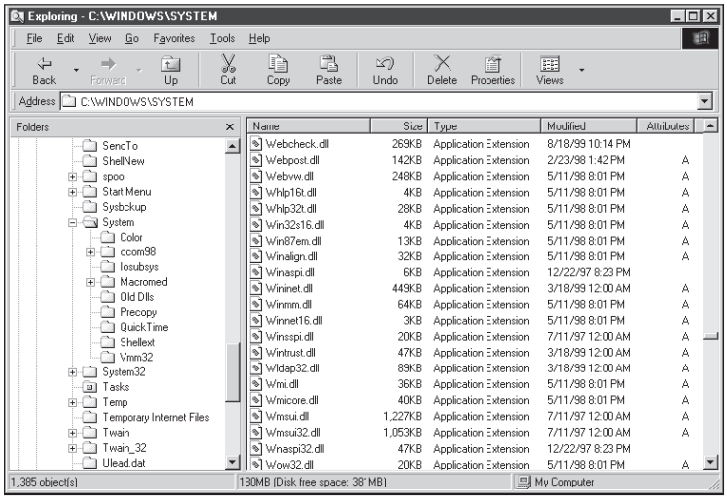


Figure 12-31 Some DLL files in the \Windows\System directory

Problems arise when an application already installed on the PC uses a .dll that has been updated by the new software, and the new .dll is not compatible with the older software, or the new .dll is corrupted. Reinstalling the older software does not help because it too only updates the .dll if it finds that the .dll file in the \Windows\System directory has an older date than the date of its version.

A+ OS 3.2

Finding the .dll that is causing the problem can be like looking for the proverbial needle in a haystack. There are over 700 .dll files in the \Windows\System directory shown in Figure 12-31. It's not uncommon for a small applications software program to install as many as 50 files in this shared directory. Furthermore, when you uninstall an applications software program, the uninstall program probably does not (and should not) erase the .dlls in this shared directory. Over time the directory can get quite large, just as the one in Figure 12-31 has.

Solutions to problems with applications software trying to share .dlls and other files in this directory involve taking more control over the installation process and knowing how to use some utility software to help diagnose software problems. Here are some steps to take to avoid and address software conflicts.

Back up the \Windows\System Directory If you have the room on your hard drive, you can back up the entire \Windows\System directory before you install new software. After the software is installed, if another program has problems, the older .dlls will be available to you. Perhaps the new software can use an older .dll, and your problem is solved.

However, how do you determine which .dll is causing the problem? First, you must know which .dlls were changed by the new software, and you must know which of these .dlls the old software uses. After that, it's usually a matter of replacing each one in turn until the problem goes away. Things can get even more complicated because .dlls sometimes work as groups. For example, you must keep the four or five files in the OLE group together as a group; exchanging only one file in the group can cause an error.

Monitor the Files Being Updated During the Installation Process Know what files an application setup program has written to the \Windows\System or any other directory and know the changes made to the Windows .ini files. Some tools you can use for this purpose are Norton Utilities, a shareware program called In Control, and a utility program called DLLaGator.

Windows 2000 has solved the problems caused by applications sharing .dll files by keeping all the files, including the .dll files, that are installed by each application in a separate folder.

Know What .dlls an Application Uses Applications load .dlls into memory and unload them throughout installation and during operation. One utility program that lets you view the .dll files currently loaded is DLL Show by Gregory Braun (*www.gregorybraun.com*). See Figure 12-32.

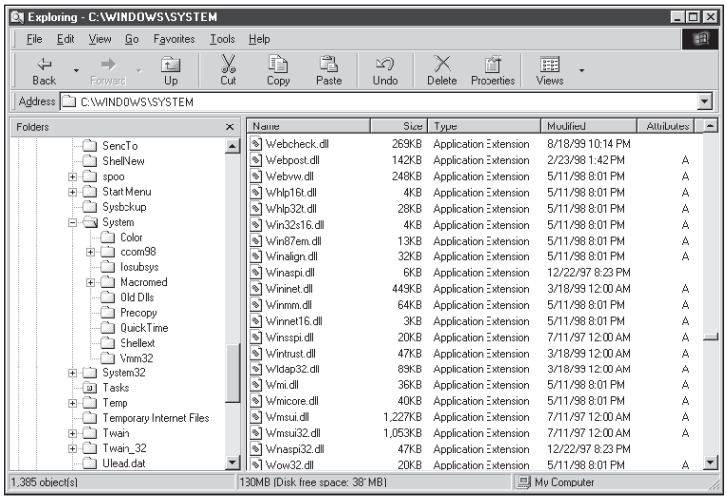


Figure 12-32 Use DLL Show to view .dll files currently in use

Run DLL Show while an application program is loaded. For example, if an applications software program is giving problems, use DLL Show to list the .dlls that might be corrupted. If the software works successfully on one PC, but generates errors on another PC, run DLL Show on the good PC. At the point where the software hangs on the other PC, look for a .dll that is loaded at that time. If you discover one, you probably have identified the file causing the problem.

DLL Show can be a very useful utility to help identify a .dll that has been overwritten by another software installation and is causing problems for the current software. However, DLL Show might not tell the entire story. Sometimes when software loads a .dll, it first reads initialization information from another file in the \Windows\System directory, which might be causing the problem. The file is used by the software but is not listed by DLL Show.

Also, when you see a Windows error message on the screen, if the dialog box offers a Details command button, click the button for more information. Sometimes the name of the .dll causing the problem is listed.

Supporting DOS Applications Under Windows 9x

A+ OS 3.2, 3.1

Windows 3.x used **PIF (program information file)** files to manage the virtual machine environment provided for DOS applications and provided a PIF editor to alter these files. Each application had its own PIF file that was used to specify the DOS environment that Windows 3.x created for it. If an application had no PIF file, Windows 3.x used the settings in the _Default.pif file in the \Windows\System directory.

Windows 9x manages the DOS box environment for DOS applications in a similar, but slightly different fashion. Apps.inf contains a section named [PIF95] that contains a master list of settings to be used for all DOS applications listed in the file.

A+ OS 3.2, 3.1

If you want to customize the settings for a DOS application, use the Properties feature of the DOS program file, which will create an individual PIF for the program file and serves as the PIF editor. Right-click the program filename and select Properties from the menu that is displayed. Windows searches for the program's PIF file and, if none is found, creates one using default values. If Windows 9x was installed over Windows 3.x, then _Default.pif still exists in the \Windows\System directory and default values are read from it. Regardless of where the default values come from, any changes made are stored in the PIF for the application.

For example, if you want a DOS program to run in DOS mode (real mode), follow these directions.

1. Right-click the program filename in Explorer and select **Properties**.
2. Click the **Program** tab and then click **Advanced**. (*Note*: The Program tab will not be present for Windows applications.)
3. Select the **MS-DOS mode** check box, as seen in Figure 12-33.

If you select Use current MS-DOS configuration, Windows executes the contents of Dosstart.bat stored in the Windows folder which is a type of Autoexec.bat file that is executed in two situations: when your select Restart the computer in MS-DOS mode from the shutdown menu or you run a program in MS-DOS mode. This file can be used to load real mode device drivers but SET commands are not executed.

If you select Specify a new MS-DOS configuration, you can make changes to Autoexec.bat and Config.sys files used for this DOS mode only. For example, since the faster 32-bit VCACHE will not be running, you can speed up disk access by loading SmartDrive from this window.

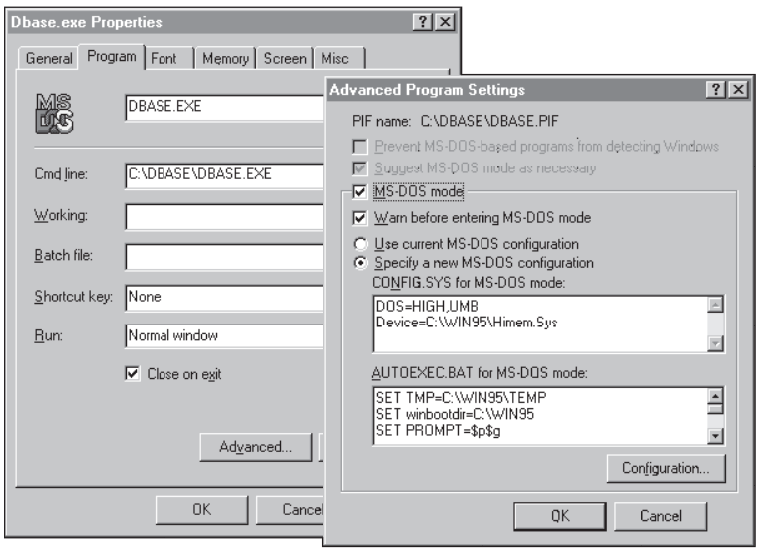


Figure 12-33 Properties sheets for a DOS application affect the way Windows 9x provides an environment for the application

Uninstalling Software

Uninstalling software is clean and easy if the software comes with a well-written uninstall program. Use the Add/Remove Program utility in Control Panel to uninstall it. However, some uninstall programs are not all that great, and sometimes a user will simply delete the folders that contain the software leaving behind .dll files in the \Windows\System directory, entries in the registry, shortcuts on the desktop, and so on for you, the PC technician, to clean up. Even worse, the user or an uninstall program might delete a .dll file needed by another application or make a wrong deletion in the registry.

When software is installed, it can do these things:

- Create new folders that belong to the application only and store files there
- Store files only used by the application in folders also used by other software. An example is an application's .ini file stored in the \Windows folder.
- Create or overwrite files used by other software. An example is \Windows\system\CTL3D.DLL, a file used by several applications that might be updated by a software installation.
- Make changes to the Windows registry
- Make changes to .ini files that belong to Windows

When software is uninstalled, deleting folders and their contents that belong only to the application is safe. Deleting entire sections in Windows .ini files that are named after the software or a branch in the registry tree that contains the application's name is also safe. But problems might occur when a change is made to a registry entry that other software depends on or when files are deleted that are used by other software.

Figure 12-34 shows the results of such an error. The problem, in this case, can be resolved by reinstalling the OCR software.

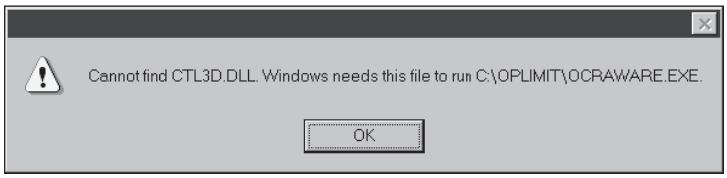


Figure 12-34 The results of an uninstall program deleting a file needed by another application

Monitoring System Performance

A+ OS 3.2

Windows 9x offers several tools listed in Table 12-8 to monitor and improve system performance and to help with troubleshooting. Some of these tools have already been discussed and some are covered next. Several of them are only available with Windows 98 and are accessed from the Microsoft System Information window. To access this window, click

A+ OS 3.2

Start, point to **Programs**, **Accessories**, **System Tools**, and click **System Information**. The dialog box illustrated Figure 12-22 opens. The System Information utility is available under Windows 95, but does not include the tools listed in Table 12-8.

Table 12-8 Tools used for troubleshooting and to monitor and improve system performance

Tool	Win95	Win 98	Description
Automatic Skip Driver Agent File name: Asd.exe Location: \windows		X	Automatically skips drivers that prevent Windows from loading and records problems encountered in the log file Asd.log. To run, select **Automatic Skip Driver Agent** from the **Tools** menu of the **System Information** window.
Microsoft System Information File name: MSInfo32.exe Location: \program files\common files\microsoft shared\msinfo	X	X	Displays system information, including installed hardware and device drivers. To run, click **Start**, **Programs**, **Accessories**, **System Tools**, **System Information**.
Hardware Diagnostic tool (Hwinfo.exe)		X	Displays the same information as System Information, but in text form. Enter **hwinfo/ui** in the Run dialog box.
Registry Checker File name: Scanreg.exe Location: \windows\command		X	Backs up, verifies, and recovers the Registry. To run, select **Registry Checker** from the **Tools** menu of the **System Information** window.
Windows Update File name: Iexplore.exe Location: *http://www.microsoft.com/windows update*	X	X	Download service packs (fixes) for Windows from the Microsoft web site.
System options in Control Panel	X	X	Several options in Control Panel can be used in monitoring and tweaking system performance.
System Configuration Utility Filename: MsConfig Location: \windows\system		X	Allows you to temporarily modify the system configuration to help with troubleshooting. To run, select **System Configuration Utility** from the menu **Tools** of the **System Information** window.
System File Checker File name: Sfc.exe Location: \windows\system		X	Verifies system files. This tool scans for changed, deleted, or corrupted system files and restores them from the originals on the Windows CD-ROM. To run, select **System File Checker** from the Tools menu of the **System Information** window.
System Monitor File name: Sysmon.exe Location: \windows		X	System Monitor tracks the performance of some important system components. To run, click **Start**, **Programs**, **Accessories**, **System Tools**, **System Monitor**.
Microsoft Backup (\program files\accessories\backup\msbackup.exe)	X	X	Backs up files and folders to prevent loss when your hard drive fails. To run, click **Start**, **Programs**, **Accessories**, **System Tools**, **Backup**.
System Recovery File name: pcrestor.bat Location: On the Windows 98 CD in \tools\sysrec		X	Uses a full system backup created by Microsoft Backup to reinstall Windows and restore the system to its state as of the last backup.
Dr. Watson File name: Drwatson.exe Location: \windows		X	Traps errors in log files created by applications and takes a snapshot of the system to use for troubleshooting.
Scheduled Task Wizard File name: Mstask.exe Location: \windows\system	X	X	Schedule tasks such as MS Backup to run at predetermined times.

A+ OS 3.2

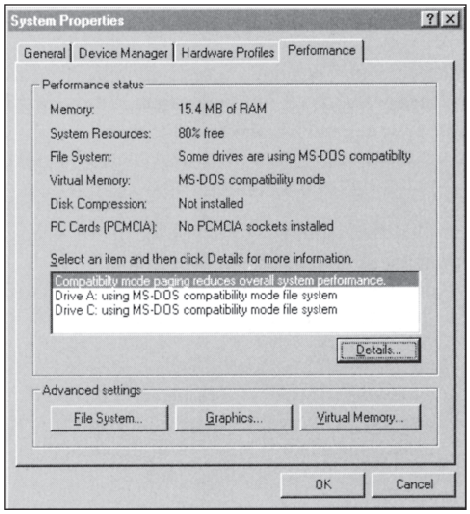
Table 12-8 Tools used for troubleshooting and to monitor and improve system performance (continued)

Tool	Win95	Win 98	Description
Version Conflict Manager File name: Vcmui.exe Location: \windows		X	Installs Windows files over a newer file that might be in the \Windows folder and subfolders
System Configuration Editor File name: sysedit.exe Location: \windows\system	X	X	Text editor to edit files that configure how Windows loads. To run it, enter sysedit.exe in the Run dialog box. Sysedit automatically opens protocol.ini, system.ini, win.ini, config.sys and autoexec.bat for editing.
Signature Verification Tool File name: sigverif.exe Location: \windows		X	Checks system drivers for digital signatures given them by Microsoft, which ensures they have been tested by Microsoft. To run it, use the System Information window.
Digital Signature Check		X	Identifies drivers that have been digitally signed by Microsoft to verify their integrity. To use it, enabled this key in the registry: HKEY_LOCAL_MACHINE\Software\Microsoft\Driver Signing

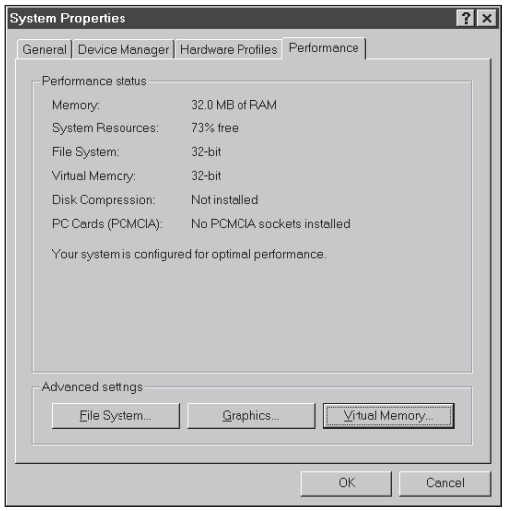
System Options in Control Panel

A+ OS 3.1

In Control Panel, select **System** and click the **Performance** tab. Figure 12-35 shows the results of this action on two computers, one in need of performance tuning and one running at optimal performance. Key messages to look for on this screen (see Figure 12-35a) are "Some drives are using MS-DOS compatibility" under File System, and "MS-DOS compatibility mode" under Virtual Memory. These messages mean that real-mode drivers are being used, which can slow down performance, especially when used with hard drive access. Figure 12-35b indicates that both these components are using 32-bit protected-mode drivers.



a. Adjustments are needed in order to use 32-bit protected-mode resource



b. System running at optimal performance

Figure 12-35 The Performance tab under System Properties in Control Panel can tell you if your file system and virtual memory are running at optimal performance

A+ OS 3.1

Whenever you see MS-DOS mode (real mode) being used, make the effort to do whatever you can to see that these drivers are replaced with 32-bit protected-mode drivers. One important tool to use for this process is the file Ios.ini, a text file that contains the Windows 9x Safe Driver List. Windows 9x uses this list to determine if it can safely substitute a protected-mode driver for a real-mode one. Also, if it attempts to make the substitution, but fails, it often records the problem in another file, Ios.log. Check this file for information about the problem.

If a real-mode driver is being used, and you believe that a protected mode driver should be used in its place, first check Ios.log for any error messages. If you don't find an error message, add the real-mode driver name to the safe driver list in the Ios.ini file. Anything following the semicolon on the line is a comment. Sample lines in the file are:

```
[SafeList]
386max.sys     ;     Qualitas
extrados.pro   ;     Qualitas Memory Manager
extrados.max   ;     Qualitas Memory Manager
4dos.com       ;     4DOS shell program
ad-dos.com     ;     Afterdark
ad-wrap.com    ;     Afterdark
adi2.com       ;     Afterdark
aspi3x70.sys   ;     DTC SCSI driver
```

If you are using third-party disk compression software, such as Stacker, make sure to use a 32-bit version of the software. When converting from Windows 3.x to Windows 9x, also upgrade Stacker. If you are still using a 16-bit version of the software, most often an error message about the problem can be found in Ios.log.

System Monitor

A+ OS 3.2

System Monitor allows you to monitor how system resources are being used by applications. It can monitor the file system, memory, the kernel, printer sharing services, and network performance data. System Monitor is not automatically installed in a typical installation. To install it, go to **Control Panel, Add/Remove Programs**. Click **Windows Setup**, and then select **Accessories**. To run System Monitor, click **Start**, point to **Programs**, **Accessories**, **System Tools**, and click **System Monitor**.

Figure 12-36 shows System Monitor tracking the kernel and disk cache hits and misses. Under the File menu you can add and delete items the monitor is tracking. Use System Monitor to help determine if an application is using an inordinate amount of resources or has a memory leak.

A+ OS 3.2

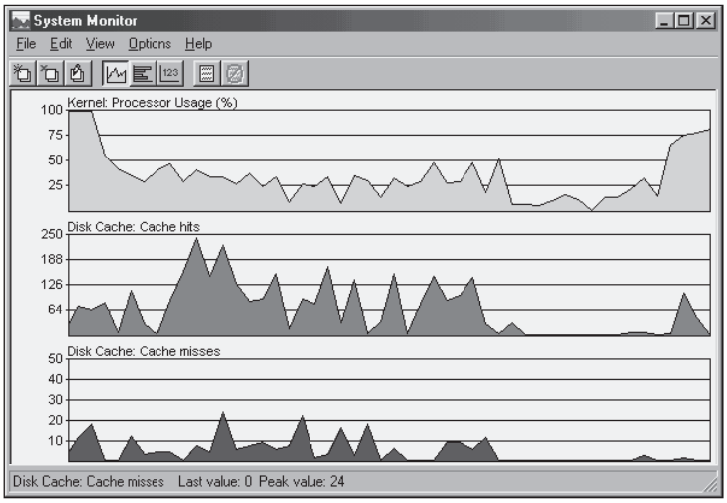


Figure 12-36 System Monitor can track the performance of several system resources

System Configuration Utility

A+ OS 3.2

Similar to loading Windows in safe mode, the System Configuration Utility reduces the startup up process to its essentials. If starting Windows in this condition eliminates the problem, then you can use this utility to add items back one at a time until the problem occurs; the problem source is related to the last item you added. To use the utility, do the following:

1. To access the utility, click **Start**, point to **Programs**, **Accessories**, **System Tools**, and then click **System Information**. The Microsoft System Information window opens (see Figure 12-22). Another way to access the Window is to type MSCONFIG in the RUN dialog box.
2. From the **Tools** menu, select **System Configuration Utility**. The System Configuration Utility dialog box opens, as in Figure 12-37.
3. To diagnose a problem, select **Diagnostic startup – interactively load device drivers and software**, and then click **OK** to restart your computer.
4. If this solves the problem, then the clean start was successful. Next, select **Selective startup** from the screen shown in Figure 12-37 and methodically select first one item and then another to restore, until the problem reappears. Begin by restoring all entries in AUTOEXEC.BAT and CONFIG.SYS, to determine if real-mode drivers and programs loaded from these files are the source of the problem.
5. If the problem still occurs, even with the clean boot, then try these things:
   - Scan for a virus, using a current version of antivirus software.
   - Use Registry Checker to check for corrupted system files.
   - Use System File Checker to check for corrupted system files.
   - Check the CMOS setup screen for wrong settings.

A+ OS 3.2

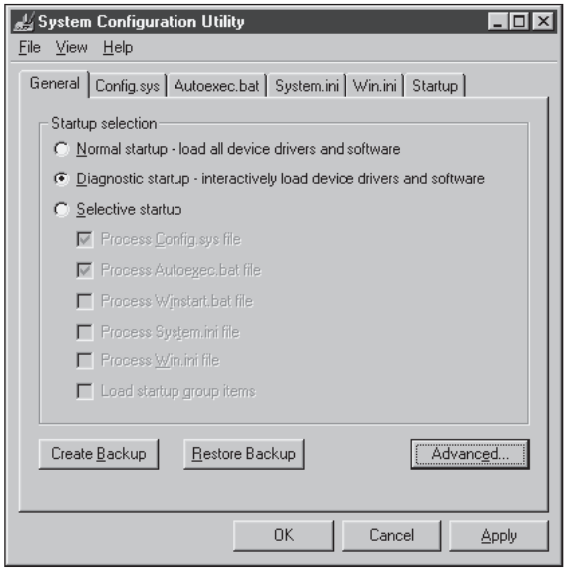


Figure 12-37 The Windows 98 System Configuration Utility helps troubleshoot Windows configuration problems

WINDOWS ME

Recall that Windows Me is the latest Windows 9x upgrade. Now that you have learned about Windows 95 and Windows 98, you can turn your attention to the features unique to Windows Me that are important if you are supporting the OS. These features include components to protect the system from failure and changes made to the OS that alter the procedures used for troubleshooting.

The Windows Me System Restore component automatically backs up the registry and other system files when the system is idle, at about every 10 hours of operating time. System Restore also prevents a user from deleting important Windows system files. If the system fails but you can still boot into Safe Mode, use the System Restore wizard. You can then choose between earlier versions of the saved system. The damaged system files will be overwritten with these saved ones.

Another feature, System File Protection, which is similar to Windows 2000 System File Protection, prevents system files from being deleted. For example, if you attempt to delete files in the Program Files folder where applications are normally stored, the utility will work in the background to compress and save these files in case you need them later. The utility also prevents an application installation from overwriting newer DLL files with older or non-standard versions.

Windows Me desktop looks more like Windows 2000 than Windows 98 SE, and, just like Windows 2000, the option to boot to the DOS command prompt is not included on the Start menu. That means that you cannot boot Windows Me from the hard drive in true real mode although you do have a MS-DOS Command Prompt window. If you want to get a

true real-mode command prompt using Windows Me, create a blank formatted floppy disk, and copy Io.sys and Command.com from the \Windows\Command\EBD folder to the disk, and then boot from this disk.

Windows Me does not allow real-mode device drivers and TSRs to be loaded from Config.sys and Autoexec.bat as does earlier versions of Windows 9x. If you want to run these 16-bit programs, your only option is to load them after Windows Me loads. One way to do this is to include them in a batch file that is listed in the Properties tab of the MS-DOS Command Prompt shortcut.

SUPPORT FROM MICROSOFT

Microsoft offers some excellent support for its products. For those serious about learning to provide professional support for Windows 95 or Windows 98, I highly recommend these books: *Microsoft Windows 95 Resource Kit* and *Microsoft Windows 98 Resource Kit*, both by Microsoft Press.

Another valuable source of information, including software utilities, enhancements, and troubleshooting guidelines for Windows 3.x and Windows 9x is the Microsoft web site. For Microsoft Technical Support Knowledge Base, access this web site:

http://support.microsoft.com/search

Figure 12-38 shows the beginning query screen for this web site. Follow the steps given there to research specific or general topics for the Microsoft products. For example, if you want to learn more about how to optimize Windows 95, choose **Windows 95** in Step 1 of Figure 12-38, and enter **optimize** as the key word to search for. Click **GO**. A list of related articles is displayed. Double-click the article to display it.

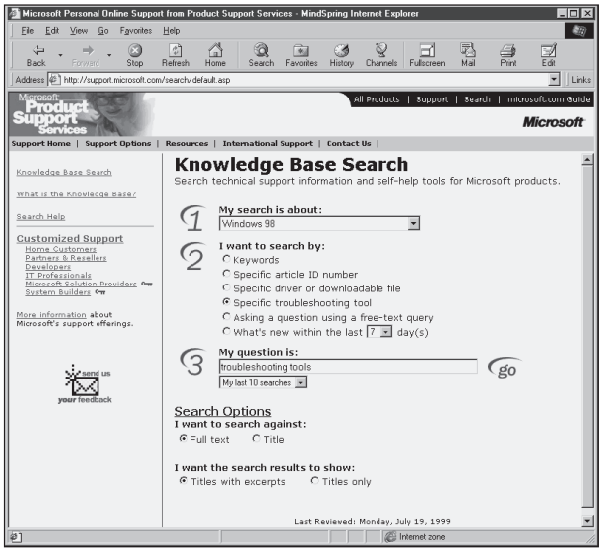


Figure 12-38 Microsoft Technical Support web site

CHAPTER SUMMARY

- Windows 9x, Windows 2000, and Windows NT are the most popular operating systems for new single-user PCs, although DOS and Windows 3.x are still used and supported.
- Software stores configuration information in initialization files with an .ini extension that are organized into sections, key names, and values, and can be edited using a text editor.
- Lines in .ini files that begin with a semicolon are comments and are ignored by the OS.
- Before upgrading Windows, scan and fix the hard drive by removing lost clusters, defragment the drive, and back up Autoexec.bat, Config.sys, System.ini, Win.ini, and the Windows registry files.
- Applications software can cause conflicts when, during an installation, the new application overwrites an existing .dll file. The new .dll file might not work correctly with previously installed applications. The .dll files are usually located in the Windows\System directory.
- Utility software can track changes to the Windows\System directory and changes to Windows .ini files and the registry during software installation.
- PCs can operate in either real mode or protected mode. Real mode limits programs to the first 1 MB of memory, allows direct access to I/O devices, and use a 16-bit data path. Protected mode gives a program access to memory addresses above 1 MB, prevents direct access to I/O devices, and uses a 32-bit data path. Protected mode is generally preferred over real mode because it is faster.
- At one time, a General Protection Fault (GPF) error always meant a memory violation by software, but now it can represent many different software errors.
- Windows application programs can give insufficient memory errors because a portion of a memory heap is not available, even though not all physical RAM is used yet.
- A memory leak is caused when an applications software program does not release a group of memory addresses back to the heap when it unloads.
- Windows 3.x is mostly a 16-bit application, whereas Windows 9x is a mixture of 16-bit and 32-bit code.
- The core components of Windows 9x are the kernel, the user, and the GDI processes.
- Windows 9x, as well as Windows NT and Windows 2000, uses the virtual machine concept to protect against program faults from software currently running.
- Memory paging is a Windows 9x method of allocating a different set of memory addresses to different virtual machines.
- Windows 9x can be customized by entries in the text file Msdos.sys.
- When Windows 9x starts up, static VxDs are loaded in real mode and then the OS switches to protected mode, in which dynamic VxDs are loaded.
- Plug and Play requires the use of 32-bit dynamic VxDs.

- Windows 9x and Windows 3.x can coexist on the same PC; this is called a dual boot.
- Press F8 when Windows 9x is loading to view and use the Windows 9x Startup Menu, which can be helpful in troubleshooting Windows problems.
- Plug and Play (PnP) is a group of architectural standards designed to automate the installation of new hardware devices on PCs.
- When using Windows 9x, in order for a PC to be completely Plug and Play-compliant, the system BIOS, all hardware devices, and the OS must be Plug and Play.
- The four components of the OS portion of Plug and Play are the configuration manager, the hardware tree, the bus enumerator, and the resource arbitrator.
- A legacy card is an expansion card that is not PnP-compliant and must be manually configured.
- Windows 9x uses 32-bit drivers stored in extended memory, although it does support older 16-bit drivers stored in the first MB of memory.
- Windows 9x loads older 16-bit drivers from either CONFIG.SYS or AUTOEXEC.BAT.
- The Windows 9x registry keeps information that was previously kept in Windows 3.x .ini files.
- The Windows 9x registry uses six major branches, or keys.
- The Microsoft web site, *support.microsoft.com*, is an excellent source of troubleshooting information for Windows.
- Useful Windows troubleshooting utilities include Automatic Skip Driver Agent, Microsoft System Information, Registry Checker, System options in Control Panel, System Configuration Utility, System File Checker, and System Monitor.

KEY TERMS

Bus enumerator — A component of Windows 9x Plug and Play that locates all devices on a particular bus and inventories the resource requirements for these devices.

Comment lines — Documentation lines that are ignored by a program. A REM in front of a line will comment out an AUTOEXEC command. A semicolon will turn an .ini file line into a comment.

Configuration manager — A component of Windows 9x Plug and Play that controls the configuration process of all devices and communicates these configurations to the devices.

Disk thrashing — A condition that results when the hard drive is excessively used for virtual memory because RAM is full. It dramatically slows down processing and can cause premature hard drive failure.

DLL (dynamic-link library) — A file with a .dll file extension that contains a library of programming routines used by programs to perform common tasks.

Dr. Watson — A Windows utility that tracks detailed information into a log file about a failed event and the software that caused it.

Dual boot — The ability to boot using two or more different operating systems, such as Windows NT and Windows 98. However, programs are not easily shared between Windows NT and the other OSs.

Dynamic VxD — A VxD that is loaded and unloaded from memory as needed.

Group files — Windows 3.x files with the .grp file extension that contain information about a program group of Program Manager.

Hardware tree — A database built each time Windows 9x starts up that contains a list of installed components and the resources they use.

Initialization files — Configuration information files for Windows. Win.ini and System.ini are the two most important Windows 3.x initialization files.

Kernel — Core portion of an operating system that loads applications and manages files, memory, and other resources.

Keys — In Windows 9x, section names of the Windows 9x registry.

Legacy — An older device or adapter card that does not support Plug and Play, and might have to be manually configured through jumpers or DIP switches.

Memory leak — A problem caused when an application does not release the memory addresses assigned to it when it unloads, causing the memory heaps to have less and less memory for new applications.

Memory paging — In Windows 9x, swapping blocks of RAM memory to an area of the hard drive to serve as virtual memory when RAM memory is low.

MSDOS.SYS — In DOS, a program file that contains part of the DOS kernel and controls much of the boot process. In Windows 9x, a text file that contains settings used by Io.sys during booting.

Object linking — A method whereby one application can execute a command on an object created by another application.

Page fault — An OS interrupt that occurs when the OS is forced to access the hard drive to satisfy the demand for virtual memory.

Page-in — The process in which the memory manager goes to the hard drive to return the data from a swap file to RAM.

Page-out — The process in which, when RAM is full, the memory manager takes a page and moves it to the swap file.

PIF (program information file) — A file with a .pif file extension that is used by an OS to store the settings of the environment provided to a DOS application.

Plug and Play BIOS — Basic input/output system for Plug and Play devices, which are designed to be automatically recognized by the computer when they are installed.

registry — A database used by Windows to store hardware and software configuration information, user preferences, and setup information. Use Regedit.exe to edit the registry.

Resource arbitrator — A PnP component that decides which resources are assigned to which devices.

Resource management — The process of allocating resources to devices at startup.

Run-time configuration — A PnP ongoing process that monitors changes in system devices, such as the removal of a PC Card on a notebook computer or the docking of a notebook computer to a docking station.

Safe mode — The mode in which Windows 9x is loaded with minimum configuration and drivers in order to allow the correction of system errors. To enter safe mode, press F5 or F8 when "Starting Windows 95/98" is displayed.

Static VxD — A VxD that is loaded into memory at startup and remains there for the entire OS session.

Value data — In Windows 9x, the name and value of a setting in the Registry.

Virtual machines (VM) — Multiple logical machines created within one physical machine by Windows, allowing applications to make serious errors within one logical machine without disturbing other programs and parts of the system.

Virtual memory manager — A Windows 9x program that controls the page table, swapping 4K pages in and out of physical RAM to and from the hard drive.

Review Questions

1. What Microsoft operating system introduced 32-bit code and protected mode into the OS?
2. Which Microsoft operating systems support 32-bit programs? Which OSs do not allow 32-bit programs?
3. Which Microsoft operating systems do not allow an application program to run in real mode?
4. Which portion of an OS is most likely to interact with the user, the kernel, or the shell? Which is most likely to interact with secondary storage?
5. If Windows runs low on memory while graphics and printing or both are being done, what heap is most likely the problem? What is the solution to the problem?
6. What is the keystroke to move from one program to another program running in a different window?
7. Name two editors that can be used to edit .ini files.
8. What .ini file can contain VxD entries that override the VxD entries in the registry that are loaded when starting Windows 9x?
9. What symbol is used to make a line a comment line in an .ini file?
10. A Windows 9x virtual machine provides an environment for _____ (16 or 32) bit programs.
11. What situation can cause a page fault?
12. What two sections in System.ini have the most impact on the boot process?
13. What is the default path to the Windows temporary directory?
14. How can you change the path to the Windows temporary directory?
15. What are the two files that make up the Windows 9x registry? What type of files in Windows 3.x does the Windows 9x registry replace?

16. What is the name of the Windows file that loads VxD drivers each time Windows starts?
17. Windows 9x uses _____ multitasking to provide backward-compatibility with 16-bit applications, but uses _____ multitasking with 32-bit applications.
18. What is the name of the Windows system file that is responsible for the switchover from real mode to protected mode?
19. In what directory are .dll files most often found?
20. What is the name of the file that contains the default settings that Windows 3.x uses to provide an environment for a DOS application?
21. What function key do you press to cause Windows to load in Safe Mode?
22. What is the keystroke shortcut to close a program in Windows?
23. In Windows 9x, what is one reason the user core uses 16-bit code rather than 32-bit code?
24. In Windows 9x, what is one reason the GDI core uses a mix of 16-bit and 32-bit code?
25. In Windows 9x, why do you think the kernel uses mostly 32-bit code?
26. What is a significant benefit to device driver developers when using the Win32 driver model (WDM)?
27. Describe the general difference between a 16-bit DOS application and a 16-bit Windows application.
28. What is disk thrashing? What causes it? What can you do about it?
29. Name three enhancements of Windows 98 over Windows 95.
30. What is the advantage of using FAT32 instead of FAT16 on a hard drive?
31. Static VxDs are loaded in _____ mode, and dynamic VxDs are loaded in _____ mode.
32. How do you force the Startup Menu of Windows 9x to be displayed during booting?
33. Using Device Manager of Windows 9x on your home or lab computer, answer these questions:
    - What are the driver files used by your video system?
    - What I/O addresses does the PCI bus use?
    - What is the name of the driver file that Windows is using to manage the PCI bus?

PROJECTS

Examine Your PC

1. List all of the .ini files stored in your home or lab computer's Windows directory.
2. Print out your home or lab computer's System.ini file and identify each hardware component that is referenced.

3. On a Windows 9x system, click the **Start** button, then click **Run:** and type **Edit C:\Msdos.sys**. Then select **File**, **Print** and print the contents of this file.
4. Follow these steps to list Windows troubleshooting tools:
   a. Click the Windows 9x **Start** button.
   b. Choose the **Settings** option.
   c. Select **Control Panel**.
   d. List at least six Control Panel utilities that can be used to configure hardware and resolve hardware problems.

Windows 9x Start Menu

As soon as your computer displays the message "Starting Windows 95/98" during the boot process, press the F8 function key. Select **Logged(\Bootlog.txt)**. When done, open the file named Bootlog.txt and print out its contents. Shut down Windows, reboot the computer, and press F8 again. Select the **Safe mode** option and note the differences in the screen's appearance. Shut down Windows, reboot the computer, and press F8 again. This time choose the **Step-by-step confirmation** option. Write down each command that executes.

Backing Up Critical Windows 9x Files

Keep a copy of critical Windows 9x files in a separate directory. Each time you install or uninstall hardware or software, redo the backup.

1. Using Windows 9x Explorer, create a new folder called Win-bak.ini.
2. In the C:\Windows folder highlight the System.ini and Win.ini files. Copy them by pressing **Ctrl+C**.
3. Click the **Win-bak.ini** directory, then paste the copied .ini files into the Win-bak.ini file by pressing **Ctrl+V**.
4. Using Explorer, set View, Options to **Show all files**.
5. Copy C:\Windows\System.dat to C:\ Win-bak.ini (do not drag!).
6. Copy C:\Windows\User.dat to C:\ Win-bak.ini (do not drag!).

Monitor a Software Installation

Using utility software such as Norton Utilities, monitor the installation of an applications software package. Install the software following the installation instructions documented with the software. Does the software allow you to customize the environment? What files are used to store the custom configuration? By looking at the date and time stamp on files, how can you guess which files contain custom setup information for this installation of the application?

Customizing a Dual Boot

You have your PC set for a dual boot between DOS with Windows 3.1 and Windows 9x. You boot to your previous version of DOS and decide to make a change to the Windows 9x Msdos.sys file. What is the current name of the file? List the steps to edit the file. Verify that your answers are correct by changing the BootMenu option in Msdos.sys. Be certain to first back up the file.

Use Control Panel to Test Your Modem

1. Click the Windows 9x **Start** button.
2. Choose the **Settings** option.
3. Select **Control Panel**.
4. Double-click the **Modems** icon. (This icon only appears when a modem is installed.)
5. Choose the **Diagnostics** tab.
6. Select the appropriate COM port and click the **More Info** button to verify that an OK message is displayed.
7. Click the **Help** button and list at least four problems that the Modem Troubleshooter will help solve.

Mouse Troubleshooting

Assume that your friend is having a difficult time finding and following the mouse pointer on his new laptop computer. Enter Control Panel and list the steps that you would use to make the mouse pointer easier to see. Test your steps on your home or lab computer.

Customizing Windows 9x

Change the Windows 9x logo screen to another .bmp file at startup. (*Hint*: See "Logo" in Table 12-5)

Using Nuts & Bolts to View the Windows 9x Registry

Nuts & Bolts offers a powerful editor for the Windows 9x registry. Use it here to save the registry to a text file (called exporting a registry file) for easy and safe viewing. This text file can later be edited and then imported back into the registry. registry text files are also convenient for transporting portions of a registry from one PC to another. In this example, you export the mouse options set in Control Panel to a registry file and view the text file.

1. Using Windows 9x, click **Start**, **Programs**, **Nuts & Bolts**, and select **Registry Pro** from the Nuts & Bolts utilities.
2. Click **Search**, **Find** and search for the text "mousekeys." The occurrences of mousekeys will be displayed in the key list at the bottom of the editor. Double-click the **HKEY_CURRENT_USER** occurrence of mousekeys. This key should now appear in the editor view window.

3. On the left side of the editor window, you can view the value names and their values for the mouse. Click **File**, **Export Registry File** to save this key to a text file. In the Save As dialog box that is displayed, enter the path and filename for the file and then click **Save**. For example, to save the file to a folder named Data, using MouseSave as the filename, enter **\Data\MouseSave**, and then click **Save**. The editor will assign a .reg file extension to the file.
4. Leaving the Registry Pro editor still on screen, use any text editor or word processor to open the file MouseSave.reg. Print the contents of this text file and compare them to the MouseKeys key values showing in Registry Pro.
5. Exit Registry Pro.

Using Nuts & Bolts Registry Wizard

Use the Registry Wizard of Nuts & Bolts to first make a backup of the registry and then perform the clean, repair, and tune-up procedures shown in Figure 12-39.

1. Back up the registry.
2. Clean the registry of the Recent Docs List. Verify the clean was successful by clicking on **Start**, **Documents**.
3. Repair the registry. Allow the Registry Wizard to search for and fix any orphan entries (entries without associated files) in the registry that it can. How many orphans did it fix?
4. Tune-up the registry. From the Registry Wizard opening screen, select **Tune-up**. Print the Registry Wizard screen showing the values or write down the size of the System.dat and User.dat files before and after the tune-up.

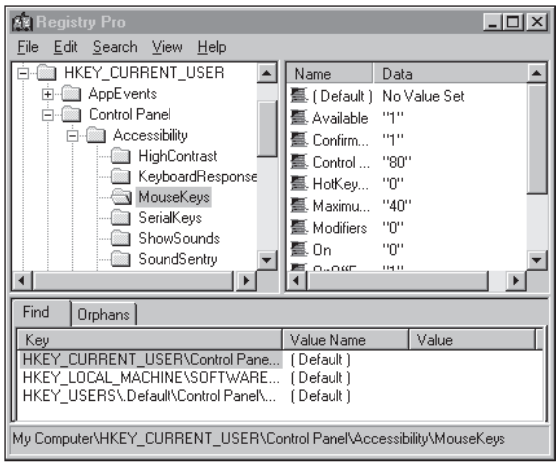


Figure 12-39 Nuts and Bolts offers a powerful registry editor, Registry Pro

Using Program Manager in Windows 9x

Program Manager is sometimes preferred by users who like the look and feel of Windows 3.x. Create a shortcut on the Windows 9x desktop to Program Manager. List the steps you took to do that.

Examine the Windows 95 Registry

Examine the Windows 95 registry to determine if your version of Windows 95 includes support for USB devices. The USB support update module is an add-on module for Windows 95 OSR 2.0. For a PC with Windows 95 to have this support, OSR 2 would have been installed and then the USB update added. To determine the installed version of the OS, look in the registry for these two values:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Version

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\VersionNumber

OSR2.0 with the USB update has the version and version number:

Version "Windows 95" and Version Number "4.03.1212" or "4.03.1214."

Print your screen showing the values for your PC. Do you have USB support installed?

Edit the Windows 9x Registry

Edit the Windows 9x registry to personalize the name of the recycle bin. Rename the recycle bin "Jean's Trash Can", using your own name. (*Hint*: Search the registry for the value "Recycle Bin".) Print the desktop screen showing the newly named recycle bin.